A Method for Dynamic Reconfiguration of a Cognitive Radio System

by

Troy Weingart

B.S., Montana State University - Bozeman, 1991

M.S., University of Colorado - Boulder, 2000

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2006

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **18 JAN 2007**	2. REPORT TYPE	3. DATES COVERED **00-00-2007 to 00-00-2007**

4. TITLE AND SUBTITLE **A Method for Dynamic Reconfiguration of a Cognitive Radio System**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of Colorado at Boulder,Boulder,CO,80309**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR’S ACRONYM(S)
	11. SPONSOR/MONITOR’S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**157**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

This thesis entitled:
A Method for Dynamic Reconfiguration of a Cognitive Radio System
written by Troy Weingart
has been approved for the Department of Computer Science

Prof. Douglas C. Sicker

Prof. Dirk Grunwald

Prof. John Bennett

Prof. Tom Lookabaugh

Prof. Dale Hatfield

Date

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Weingart, Troy (Ph.D., Computer Science)

A Method for Dynamic Reconfiguration of a Cognitive Radio System

Thesis directed by Prof. Douglas C. Sicker

Advances in process technology, manufacturing, and architecture have ushered in an age of faster, smaller, and cheaper electronic devices. Emerging processor technology has made it possible to migrate applications that were traditionally implemented in custom silicon to general purpose processors. In the area of wireless communications, this transition has given birth to the field of cognitive and software-defined radio (C/SDR). These C/SDRs offer a broad range of opportunities for improving the use and utilization of radio frequency spectrum. This includes the creation of radio networks that can reconfigure their operation based on application requirements, policy updates, environmental conditions, and the ability to adapt to a wide range of protocols. One of the key benefits of having a C/SDR is its ability to change communication parameters in response to changes in application needs and/or changes in the radio frequency landscape. Such reconfiguration requires an understanding of how these communication parameters interact within the network protocol stack. Analysis of these parametric cross-layer interactions is a critical precursor in the development of a predictive model and algorithm for dynamic reconfiguration of a C/SDR.

This work investigates how parameters at the physical, data link, network, and application layers interact, how desirable configurations of these parameters can be determined, and how these parameters affect the performance of file transfer and Voice over IP applications. An analysis of varying communication parameters across networking layers is used to inform the design, implementation, and evaluation of a predictive model and algorithm for dynamic reconfiguration of a cognitive radio. This model and algorithm allow a C/SDR to dynamically modify its configuration in order to improve system performance. A systematic method for development of a cognitive platform is presented. This method uses statistical analysis of variance and design of experiments techniques to inform the design and implementation of a dynamic reconfiguration algorithm. This algorithm exploits cross-layer interactions to improve system performance, adapt to the needs of users, and respond to changes in the radio frequency environment.

Dedication

This work is dedicated to my family. To my wife, the love of my life, faithful friend, and devoted mom; and to my kids, who were always cheerful and understanding when Dad was typing on the computer rather than reading to them or playing soccer in the park. It was a long and difficult journey for our family with the stress of the PhD and living apart from them for the better part of three years. I love you all very very much and couldn't have made it without you. Finally, I would like to thank my parents for their love, support, and encouragement through the years (love you Mom and Dad).

Acknowledgements

First and foremost I would like to thank my advisor, Doug Sicker, for his patience, tireless effort, outstanding guidance, understanding and friendship. I would also like to thank my co-advisor, Dirk Grunwald, for his thoughtful insight, willingness to offer feedback, and friendship. Finally,pysh I'd like to thank the members of the CU systems lab for their help in creating graphs, proofreading publications, and friendship over the years.

Contents

Chapter

1 Introduction 1

1.1 Challenges in Cognitive/Software Radio 6

1.2 Research Focus 10

1.3 Major Contributions 11

1.4 Overview 13

2 Review of Literature 14

2.1 Software Defined Radio 14

2.1.1 SPEAKeasy: Military Software Radio 15

2.1.2 Joint Tactical Radio System 16

2.1.3 Rapidly Deployable Radio Network 17

2.1.4 Virtual Radios/VANU Inc. 17

2.1.5 End to End Reconfiguratility (E2R) 18

2.1.6 SoftMAC and MultiMAC 18

2.1.7 Summary 19

2.2 Cognitive Radio 20

2.2.1 Genesis Of Cognitive Radio 21

2.2.2 DARPA neXt Generation Communications 22

2.2.3 Cognitive Radio Networks 24

2.2.4 Algorithms for Cognitive Radio 25

2.2.5 Summary 27

2.3 Reliability Within a Network Protocol Layer 28

2.3.1 Physical Layer 29

2.3.2 Data Link Layer 30

2.3.3 Network Layer 31

2.3.4 Transport Layer 31

2.3.5 Summary 33

2.4 Dynamic Protocol Adaptation 33

2.4.1 Cross-layer Optimization 33

2.4.2 Flexible and Adaptive MAC Design 35

2.4.3 Summary 37

2.5 Focus of Related Work 37

3 Method 39

3.1 Assumptions 40

3.2 Problem Decomposition 41

3.2.1 Component Framework 41

3.2.2 An Abstraction for C/SDR 44

3.3 Experimentation 47

3.3.1 Layout and Setup 48

3.3.2 External Parameters and Environmental Variables 48

3.3.3 Internal Parameters 49

3.3.4 Metrics 51

3.3.5 Analysis with Average Effect of a Parameter 52

3.3.6 Analysis with Design of Experiments 53

3.4 Algorithm Development 59

3.4.1 Algorithm Decomposition . . . 59
3.4.2 Metrics for Algorithm Evaluation . . . 64
3.5 Experimental Platform - Simulation in OPNET . . . 65
3.5.1 Setup and Layout . . . 65
3.5.2 Limitations of the Simulation Platform . . . 66
3.5.3 Custom Simulation Model . . . 67

4 Simulation Results and Findings 71
4.1 Experimental Analysis . . . 71
4.1.1 Average Effect of Parameter . . . 72
4.1.2 Analysis of Variance . . . 88
4.1.3 Impact of a Parameter and Multi-Factor Analysis . . . 109
4.1.4 Experimental Analysis - Summary of Findings . . . 121
4.2 Algorithm Performance . . . 122
4.2.1 Comparative Performance . . . 123
4.2.2 General Performance . . . 132
4.2.3 Algorithm - Summary of Findings . . . 133

5 Conclusion 134
5.1 Contributions . . . 134
5.2 Findings . . . 135
5.3 Future Work . . . 135

Bibliography 137

Tables

Table

1.1 Radio Taxonomy 2

3.1 External Parameters and Environmental Variables 49

3.2 Internal Parameters 50

3.3 Metrics 52

3.4 Factors and Responses for 802.11 Wireless Test Example 56

3.5 Results from 802.11 Wireless Test Example 57

3.6 ANOVA for Latency 58

3.7 Reconfiguration Metrics 65

4.1 Internal Parameters 72

4.2 External Parameters and Environmental Variables 72

4.3 Metrics 72

4.4 Perspectives on Performance Metrics 73

4.5 ANOVA for HL Throughput 89

4.6 ANOVA for FTP Throughput 90

4.7 ANOVA for VoIP Throughput 91

4.8 ANOVA for RMAC Throughput 92

4.9 ANOVA for HL Bit Loss 94

4.10 ANOVA for FTP Bit Loss 95

4.11 ANOVA for VoIP Bit Loss 96

4.12 ANOVA for RMAC Bit Loss 97

4.13 ANOVA for HL Latency 99

4.14 ANOVA for FTP Latency 100

4.15 ANOVA for VoIP Latency 101

4.16 ANOVA for RMAC Latency 102

4.17 ANOVA for HL Jitter 104

4.18 ANOVA for FTP Jitter 105

4.19 ANOVA for VoIP Jitter 106

4.20 ANOVA for RMAC Jitter 107

4.21 Top Three Factors Influencing Response 109

4.22 Top Three Factors Influencing Response and Their Effect 120

4.23 Best and Worst Case Static Configurations 123

Figures

Figure

1.1 European Spectrum 5

1.2 Challenges in Cognitive Radio 11

2.1 Development Timeline 15

2.2 DARPA XG Architecture 23

3.1 Research Method 40

3.2 Component Framework 43

3.3 Adverbs for C/SDR 46

3.4 Experimental Process 47

3.5 Experimental Layout 48

3.6 Average Effect on VoIP Latency 53

3.7 Prediction Cycle 54

3.8 Components of the Cognitive Algorithm 60

3.9 Message Exchange for Reconfiguration Cycle 62

3.10 Configuration Selection Process 64

3.11 Simulation Layout 66

3.12 Node Model 68

3.13 Cognitive Process 69

3.14 Media Access Control Layer 70

4.1 Average Effect of a Parameter on HL Throughput 75

4.2 Average Effect of a Parameter on FTP Throughput 75

4.3 Average Effect of a Parameter on VoIP Throughput 76

4.4 Average Effect of a Parameter on RMAC Throughput 76

4.5 Average Effect of a Parameter on HL Bit Loss 78

4.6 Average Effect of a Parameter on FTP Bit Loss 78

4.7 Average Effect of a Parameter on VoIP Bit Loss 79

4.8 Average Effect of a Parameter on RMAC Bit Loss 79

4.9 Average Effect of a Parameter on HL Latency 81

4.10 Average Effect of a Parameter on FTP Latency 81

4.11 Average Effect of a Parameter on VoIP Latency 82

4.12 Average Effect of a Parameter on RMAC Latency 82

4.13 Average Effect of a Parameter on HL Jitter 84

4.14 Average Effect of a Parameter on FTP Jitter 84

4.15 Average Effect of a Parameter on VoIP Jitter 85

4.16 Average Effect of a Parameter on RMAC Jitter 85

4.17 Interaction of Data Rate and FEC on Throughput 111

4.18 Interaction of Data Rate and ARQ on Throughput 112

4.19 Interaction of FEC and ARQ on Throughput 113

4.20 Interaction of FEC and ARQ on Bit Loss 114

4.21 Interaction of FEC and xmitPower on Bit Loss 115

4.22 Interaction of ARQ and xmitPower on Bit Loss 116

4.23 Interaction of Data Rate and FEC on Latency 117

4.24 Interaction of Data Rate and ARQ on Latency 118

4.25 Interaction of FEC and ARQ on Latency 119

4.26 Interaction of Data Rate and FrameSize on Jitter 120

4.27 Reconfiguration Algorithm 122

4.28 MCC vs. Best and Worst Static Configurations on Throughput 124

4.29 MCC vs. Best and Worst Static Configurations on Bit Loss 126

4.30 MCC vs. Best and Worst Static Configurations on Latency 127

4.31 MCC vs. Best and Worst Static Configurations on Jitter 128

4.32 Radio Frequency Propagation for MCC vs Best Configuration 129

4.33 MCC vs. Best and Worst Static Configurations - Average Throughput . 130

4.34 MCC vs. Best and Worst Static Configurations - Average Bit Loss . . . 130

4.35 MCC vs. Best and Worst Static Configurations - Average Latency . . . 131

4.36 MCC vs. Best and Worst Static Configurations - Average Jitter 131

4.37 MCC vs. Best and Worst Static Configurations - Average Power Output 132

4.38 Time to Reconfigure 133

Chapter 1

Introduction

A convergence of technology trends is changing the operational and design characteristics of radio devices. The first of these technology trends is software-defined radio (SDR). SDRs allow much of what was previously done with application specific hardware, such as signal processing, to be accomplished in software. The ideal software radio is one that allows all signal processing, with exception of the analog antenna, to be done digitally [14]. Next are frequency agile devices, systems that can change how and where they operate within the radio spectrum, moving among a set of frequency bands in response to interference or other constraints. Finally, there is cognitive radio (CR), wherein the device can autonomously make decisions about its operation, in response to environmental changes, such as interference. Such capabilities enable SDRs to vary waveforms and frequencies within their hardware constraints, as well as change how their network protocols operate. One analogy that has worked well when discussing C/SDR stems from the way in which traditional analog radio devices were operated. These older devices required the turning of a knob to change frequency or the flipping of switches to enable features. A cognitive engine running on a software-radio would dynamically “turn” and/or “flip” these virtual knobs and switches to affect the radio. While this analogy breaks down in its oversimplification of the problem, some may find it useful.

A SDR can be described as a transceiver in which much of the physical layer is programmable, allowing the device to be reconfigured to meet changing needs or changes in the radio frequency (RF) environment. Key to this ability to reconfigure is support for different modulation schemes, frequency adaptation, and portable waveforms. The International Telecommunications Union Radiocommunications (ITU-R) defines an SDR as, "A radio in which the RF operating parameters of frequency range, modulation type, and/or output power can be set or altered by software, or the technique by which this is achieved [40]." Higher up the network protocol stack we have the ability to adapt the media access control and routing capabilities, although this higher layer functionality is generally not considered part of an SDR. The foundations of SDR are rooted in a collection of hardware and software whose objective is to move as much of the radios processing from the analog to the digital domain and from the hardware to the software domain, thus affording the flexibility, interoperability, and efficiency required [14].

Table 1.1: A Taxonomy of Traditional, Software-Defined and Cognitive Radios

Radio Type	Characteristics
Traditional	Specific functionalities determined during initial design Design based on non-adaptive models Implemented in hardware, e.g., ASIC-based technology Limited upgradeability
Software Defined	New functionality can be added through software updates Design largely based on software design models Implemented on general purpose processors, relies on software Enhanced upgradeability
Cognitive	Required functionality can be determined and negotiated Combines software design and reasoning techniques Implemented in flexible hardware and/or software Intelligent upgradeability

In 1999, Mitola coined the term cognitive radio (CR), classifying a CR as a device that is aware of its environment and can adapt its operation in complex manners [39]. At its core, a CR can sense, adapt and learn from its surroundings. Together, cognitive and software-defined radio (C/SDR) can combine into a flexible and aware device. This combination allows for the creation of frequency agile radios, in which the device can autonomously select operating parameters in a way that improves spectrum utilization (referred to as dynamic spectrum utilization). One can envision an advanced wireless network that dynamically assigns spectrum or reconfigures in response to changes in policy and environmental conditions. Flexible spectrum usage raises interesting issues regarding the role of primary and secondary users. For example, to what extent is a secondary user allowed to interfere with a primary user? This level of flexibility in a wireless platform not only allows one to tackle the problem of spectrum utilization, but also offers the ability for devices to cooperatively configure themselves to support application quality of service demands in a networked environment. Table 1.1 provides a summary of the different classes of radios and their characteristics.

C/SDRs have a broad range of capabilities and designs. At one end of this continuum are radios that incorporate computational intelligence. These radios require algorithms that sense, learn and act in response to changes in their environment [39]. The ultimate cognitive radio will be able to autonomously negotiate and propose entirely new communication protocols for use in a networked environment. In other words, this technology would allow devices to dynamically reconfigure their operational parameters to enhance performance metrics such as throughput, bit error rate and/or delay. At the other end of this continuum are devices that are functionally equivalent to their analog predecessors, although the newer wireless devices functionality has been implemented on field programmable gate arrays (FPGA), digital signal processors (DSP) and/or general-purpose processors (GPP).

One might appropriately ask whether frequency agile SDRs represent anything new. In some ways they do not. There are many systems and devices that incorporate some subset of C/SDR technology. For example, second generation cellular technology incorporates sophisticated transmit power control. Even typical 802.11 cards support many of the characteristics that we are seeking - they rely heavily on software to operate (although not at the physical layer); they can sense their environment (through listen-before-talk and channel sounding); and they can alter their frequency parameters (within a set of channels). However, they do all of these things in a rather rudimentary manner. One could also argue that even conventional hardware radios have some of these characteristics; however, they are much more limited in terms of their ability to dynamically alter their operation and they have certain physical limits in terms of reconfigurability (due to the constraints of adding additional hardware to support additional features). The configurations of such conventional devices are generally set during the design phase and there is little to no ability to alter this configuration post development, whereas SDRs overlay a software architecture on top of this model. This ability to alter the workings of the device post-production is powerful, in that it offers the ability to upgrade and/or patch a device that has already been deployed.

Three well known motivations for instantiating C/SDRs are (1) for purposes of efficiently accessing spectrum, which is perceived as a scarce resource and (2) for improving interoperability of wireless devices, and (3) improving economies of scale in manufacturing. In the US and abroad, numerous government agencies (e.g., the Federal Communications Commission (FCC), the National Science Foundation (NSF), the Defense Advanced Research Projects Agency (DARPA), and the European End-to-End Reconfigurability Project (E2R)), have realized that current radio technology will not adequately meet the demands of the future.

Driving a shift in both research and funding is a desire for increased efficiency in our use of the available radio frequency spectrum. A recent study done in the Atlanta

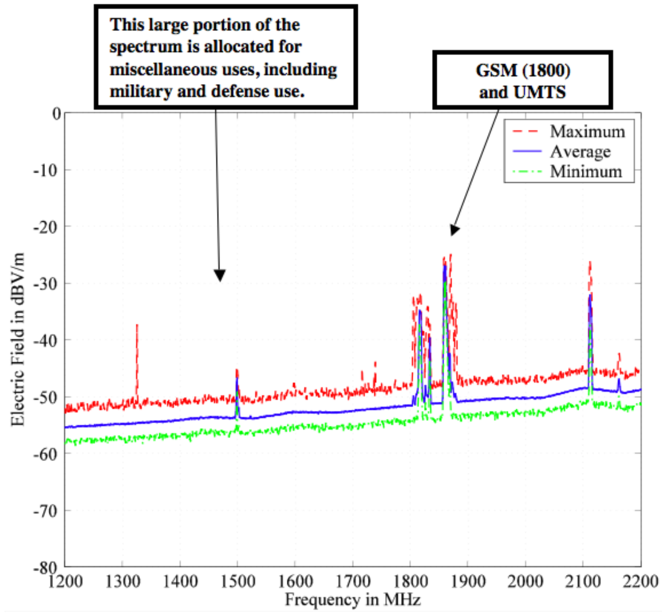

Figure 1.1: Mid-high Band Spectrum Sample from a Large City in Europe, 1,200 MHz to 2,200 MHz (This graph is courtesy of Patrick Ryan)

metropolitan area reported that less than 7% of the available spectrum was being utilized during the several months that the study was conducted. Additionally, 77% of the spectrum in this area was completely unused [64]. However, observed availability does not mean that the spectrum is not "owned" by some agency or corporate body. The scarcity of spectrum is influenced by both technical and regulatory practices. However, "ownership" issues aside, it is clear that there is a vast amount of unused and underutilized spectrum. Figure 1.1 presents a graph of spectrum occupancy from 1.2 GHz to 2.2 GHz in a large European city (note the low occupancy in the band with the exception of the UMTS and GMS services).

In addition to efficient spectrum use, there has also been a large amount of funding directed at solving interoperability problems encountered between military components and our allies during the conflicts in the Middle East. Domestically, we saw similar interoperability problems between public safety agencies responding to the 9/11 attacks. The ineffective use of the emergency bands during these terror attacks has given impetus to many of the scenarios that illustrate the promise of C/SDR. A government agency could enact a change to local spectrum policy in response to a disaster. These updates to policy, when acted upon by an C/SDR, would reallocate spectrum to support increased demand during the emergency. C/SDR also promises to change the way radio and equipment updates are done. A C/SDR platform could adapt to new standards by downloading and installing new software. A C/SDR could reduce the cost to upgrade system software through remote system updates and upgrades. There are significant economic benefits to be had if one can intelligently exploit the highly flexible and reconfigurable nature of SDR platforms. Developers of wireless hardware can rely on the reconfigurable design of an SDR to decrease the costs associated with upgrading custom hardware (such changes can result in fabrication costs that significantly exceed US $1 million).

1.1 Challenges in Cognitive/Software Radio

Challenges in C/SDR are loosely divided into two categories, those dealing with policy or regulatory concerns and those dealing with technical problems. Often researchers' interests are consumed by seemingly larger technical issues, when in reality these issues are confounded by policy and regulatory roadblocks. In order to understand the regulatory and policy issues one must understand the key desires of the regulator, primary spectrum user, and their opinion of the fundamental technical issues driving policy.

The ability of frequency agile radios to sense radio frequencies and alter operational parameters such as frequency, power, modulation, beam direction and link access protocol creates an opportunity whereby devices could autonomously decide how best to operate to improve their access to the radio frequency environment. However, assessing the radio environment is a difficult task, one that may require the development of new hardware and software, complex propagation assessment techniques and information integration. It also will require the radio community to rethink what defines harmful interference. New certification and assurance techniques will have to be developed in order to assure that frequency agile devices will not misbehave. To understand the need for certification, consider the problems that might arise if secondary user devices interfere with public safety and aeronautical users. The hope is that such adaptive devices will (1) allow for open radio architectures, (2) take advantage of price declines in computing devices, (3) support novel methods for accessing the spectrum and (4) reduce the custom nature of radio chip design. However, it is uncertain whether these devices might intentionally or inadvertently operate outside of their expected RF parameters. Regulators and incumbents must be comfortable with C/SDR technology before they will allow it to operate. Likely this will involve certification of the algorithms and mechanisms that drive the SDR. For incumbent RF license holders, certification gives assurance that the CR will not interfere in their transmission, and that the CR operating in their RF space will relinquish spectrum when required. There is also a large effort at reworking the current command-and-control model of spectrum allotment, assignment and allocation in the FCC. A transition to a legal structure that is more suited to advancements in radio technology, such as C/SDR and ultra-wideband, is underway.

Various national regulatory agencies are working on improving spectrum policy, notably the Office of Communications (Ofcom) in the UK, the Ministry of Internal Affairs and Communications (MIC) in Japan and the FCC in the US. Ofcom recently released an in depth analysis of C/SDRs and has been active in promoting novel approaches to spectrum management [88]. In the US, the FCC is actively seeking to improve the efficiency of spectrum management and therein considering new regulatory models. These include (1) the development of secondary markets for spectrum, (2) the specification of receiver standards, (3) the consideration of C/SDRs, (4) exemptions for operation in TV bands and (5) the development of an interference temperature [19, 20, 21]. This regulatory interest represents a fundamental shift in thinking, and is a key indicator that devices that use spectrum in new and exciting ways are in development. The significance of this type of regulatory reform must also be considered in the context of how regulatory agencies typically evolve. The potential for this new technology is quite substantial. However, this potential could be lost or delayed if the regulators find that the technology is not up to supporting new spectrum models. One concern for policy makers is whether interference can be avoided, and whether the device will operate as it should. The potential to interfere with primary spectrum users must be considered, particularly since these incumbent spectrum holders have substantial lobbying influence. Furthermore, current international treaties and laws do not consider spectrum sharing. Regulatory and policy issues notwithstanding, there are also a host of daunting technical problems to overcome.

While not the focus of this thesis, there are also many interesting problems at the physical layer to consider. Various physical layer advances are required to achieve highly flexible radios. One of the major challenges in creating a flexible SDR is that of developing a flexible RF front end. This challenge is being met through the ongoing development of Micro-Electro-Mechanical Systems (MEMS) based antenna [18]. Additionally, development of analog to digital converters that are appropriate for SDR systems requires changes to the sampling capability and the range of existing systems. Significant work is underway in the area of FPGA based SDRs [85]. Advances in these areas will have a significant impact on the realization of more flexible SDR platforms and therefore more flexible frequency agile radios.

Many of the technical challenges in C/SDR are driven by Joe Mitola's vision of the ultimate C/SDR [39]. He envisioned a system that was capable of reasoning about its environment. The system would draw upon past experience and current environmental conditions to make intelligent decisions about how it should reconfigure. Key components of realizing his vision are the radios ability to sense, remember, learn and act. In developing such technology, it will be important to understand how a C/SDR should sense the environment, determine what environmental inputs should be processed and when, and also how it will communicate with other network devices. Additionally, it will be import to investigate how C/SDRs synchronize configurations, maintain communication during reconfiguration, and accomplish over-the-air software updates. Changes in the environment, goals, or networking conditions will force the radio to change its current operating mode, much in the way an operating system must respond to interrupts. One must understand how tradeoffs will be balanced by the C/SDR in meeting its goals. Additionally, it will be important to define and evaluate how the C/SDR is affected by interrupts to its current operating mode. The adaptive and dynamic natures of these systems lead to other interesting technical questions. It will be important to quantify the amount of time that a cognitive process can devote to computing a radio configuration, thus allowing characterization of the types of processing can be done without negatively affecting communication. Another challenge is devising a set of metrics that can be used to evaluate and guide the design of the computational engine. Additionally, there are numerous machine-learning problems that must be resolved before realizing the C/SDR vision. One must characterize what types of reconfiguration algorithms (case-based-reasoning, genetic algorithms, expert systems) are appropriate for networks of C/SDRs. These are but a few of many problems that must be overcome in distributed systems, security, and operating systems with respect to C/SDR.

Possibilities abound for investigation into the utility of the SDR platform to capitalize on the benefits of cross-layer interaction and dynamic reconfiguration. A C/SDR could be used for research and experimentation with spatially aware applications, adaptive routing, cognitive media access control (MAC) layers, and mutable physical layers. The added flexibility and computational power afforded by a C/SDR has drawn manpower and money to their research and study.

1.2 Research Focus

Frequency agility, or the ability of the radio to dynamically alter the frequency it communicates over, is receiving a large amount of interest. The focus on frequency agility is driven by scarcity of the radio frequency resource and the potential for profit, provided one can uncover a way to improve use and utilization of that resource. Research in dynamic spectrum use is centered on ways to vacate or negotiate a clear communication band. However, there are situations in which a C/SDR cannot jump to an interference free channel to operate (i.e., the presence of military jammers, regulatory limitations, and/or scarcity of resources). This thesis focuses on what a C/SDR can accomplish to maximize its performance within a frequency band in the presence of interference. Specifically, the thesis of this work is:

> **A C/SDR system that is experiencing active interference can improve performance by exploiting dynamic cross-layer parametric optimization.**

The scope of this work is limited to those problems faced by a C/SDR when dealing with a single jamming source, two stationary nodes, and a single communication link between them.

In approaching this work, a decomposition of the problem lead to two interdependent areas of focus. For the first area, it followed that in order to develop a dynamic control algorithm for the C/SDR, one must first have a clear understanding of how the C/SDR's settings affect its performance. For example, to optimize for throughput it is important to know the effect of tuning the radio to - bit rate (high), frame size (small), and transmit power (low). Once understanding the effects of the radio's settings, one can move on to the second area of focus, the development of the algorithm. Informed design is the key principle in development of the cognitive engine. Chapter 3 will cover the details of this approach and discuss the techniques used for quantifying and analyzing the experimental results, and show how this drove the development of the algorithm.

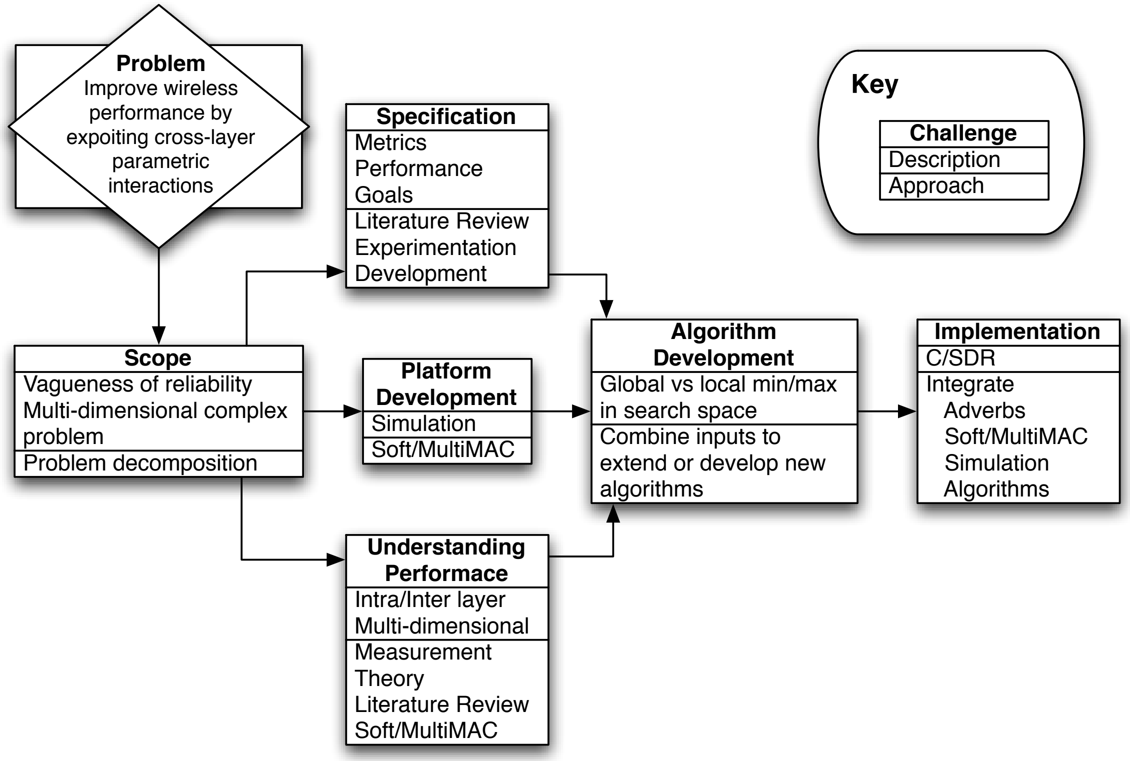

Figure 1.2: Challenges, Proposed Approaches, and Their Relationship to the Problem.

The scope of the challenges outlined in the previous section is enormous. However, when decomposed and focused by a directed thesis the tasks become more manageable. Figure 1.2 summarizes relevant challenges, the approach, and the relationship to this research. The following section will summarize the major contributions of this work.

1.3 Major Contributions

The major contributions of this research are divided into three categories, 1) the experimental parametric research, 2) the reconfiguration algorithm, and 3) the research platform. The first major contribution of this work is the presentation and analysis of the parametric interactions of the C/SDR platform. The experimental work makes use of Design of Experiments (DOE) and Analysis of Variance (ANOVA) to determine how a cognitive radio's settings affect its performance. This analysis exhaustively examines how each configuration of the C/SDR affects the performance of the system. The analysis includes single and multi-parameter interactions and effects. Additionally, a model for predicting a radio's performance given its configuration is presented. The detailed results of the experimental investigation are reported in chapter 4.

The second major contribution of the research is the design, implementation, and testing of an algorithm for dynamic reconfiguration of a cognitive system. The algorithm is based on achieving desired performance goals while maintaining the most conservative configuration (i.e., the configuration which meets requirements while maintaining minimum power output). The algorithm is discussed in detail in chapter 4.

The final major contribution of this work was the development, testing, and evaluation of a platform for experimentation with C/SDR. The simulation platform developed in OPNET offers the researcher the ability to investigate how a C/SDR's settings affect metrics of interest, validating the potential of this technique. The platform is flexible enough to allow communication parameters to be changed on a per-packet basis. Additionally, the platform allows for the development and comparison of algorithms for configuration of a C/SDR. This platform is discussed in detail in chapter 3.

During the investigation of the thesis many related research questions were explored. Unavoidably, there were tradeoffs in achieving reliable communication and adequate performance over an actively jammed link. This work presents a method for balancing these tradeoffs. How a C/SDR is affected by changes to its current environment or operating mode revealed which changes required immediate, delayed, or no action. Quantifying the amount of time that a cognitive process devotes to computing a radio configurations is presented, thereby allowing characterization of the types of processing can be done without negatively affecting communication. A set of metrics was developed to evaluate and guide the design of the C/SDR's computational engine. Finally, this work offers a method for specifying the requirements, goals, and limitations of the application or user requesting service from the network.

1.4 Overview

The remainder of this document is organized as follows. The literature review in chapter 2 describes prior work and contrasts this research with related academic and commercial work. This is followed in chapters 3 and 4 by a detailed look at the approach, platform, experimentation, and evaluation techniques that were undertaken, followed by detailed findings. Finally, this thesis concludes with a summary of the most significant results and a look at future work.

Chapter 2

Review of Literature

Cognitive and Software-Defined Radio (C/SDR) is a research area that is very broad in scope and could potentially benefit from a highly varied review of the literature. This chapter is focused primarily on methods and techniques, as described in prior work, that help achieve the focus of this thesis. Work that offers to improve performance through cross and intra-layer adaptation, interlayer adaptation, and methods for tuning the C/SDR is of particular interest. However, this chapter is purposefully broad in order to provide scope. The methods that will provide the most impact were targeted for use in the thesis of this work. Each of the following sections is devoted to an area of work directly related to this thesis. This includes a broad range of topics and aspects of SDR, CR and cross-layer and adaptive protocol design. Figure 2.1 provides a timeline of some of the major SDR developments.

2.1 Software Defined Radio

This section provides a historical overview of software defined radios and introduces two key components employed in this thesis. As described in the previous chapter, Mitola coined the term software defined radio while he was working at the Defense Advanced Research Projects Agency (DARPA). During the mid 1990s, DARPA and the Department of Defense (DOD) funded various projects crucial to the development of SDR. In recent years, academia and industry have taken the lead in advancing SDRs.

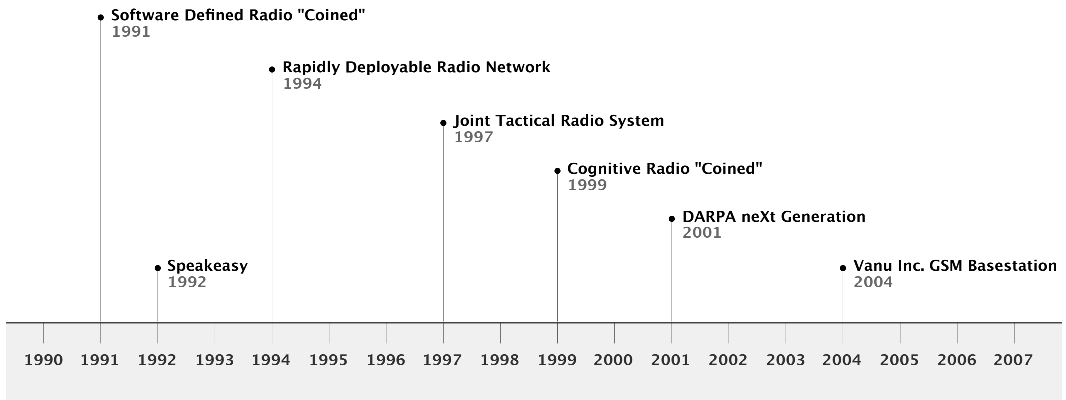

Figure 2.1: Timeline of Software-Defined and Cognitive Radio Developments

2.1.1 SPEAKeasy: Military Software Radio

SPEAKeasy was a joint program of the Air Force, Army, Navy, and DARPA that arose out of interoperability problems that were experienced during the Grenada conflict [48]. Rumor has it that Army troops called in air support using their personal calling cards. SPEAKeasy sought to develop a radio that would allow multiple waveforms at multiple frequencies to coexist in the same device and be compatible with legacy radios. Along with basic research and development, their production schedule included prototype demonstrations. The initial prototype, shown in 1995, demonstrated multi-band operation, programmability, and bridging. Development and prototype demonstrations continued until late in 1998. Some key ideas and concepts for software-defined radios arose during the development of this system. There was a need for a software architecture that would decouple the radio hardware from the software framework. This would allow hardware to advance independently of the software architecture. Additionally, they saw the need to capitalize on commercially available products, as the DOD could no longer afford to fund proprietary solutions. In order to ease maintenance of the radio, the developers saw a need for over-the-air updating of the radio software. SPEAKeasy was nominally one of the first software radios in operation. Many of the novel ideas and innovations that followed have roots in this program.

2.1.2 Joint Tactical Radio System

The Joint Tactical Radio System (JTRS) was founded on concepts that were part of the SPEAKeasy project and a vision document within the military (Joint Vision 2010/2020) [83]. This program is funding the production of a host of radios ranging from inexpensive terminals, with reduced waveform support, to multi-band/mode/channel radios that support advanced narrowband and wide-band capabilities. These high-end radios will also include integrated computer networking features. The primary goal of JTRS is to develop a family of radios that are able to meet expanding bandwidth requirements, are interoperable, affordable, and scaleable. The program managers felt that the only way to meet these objectives was to require that the radios conform to open physical and software architectures. JTRS has experienced significant difficulties in meeting program goals and cost constraints [15].

A key component of the JTRS is the Software Communications Architecture (SCA). The SCA is an open architecture framework that utilizes Common Object Request Broker Architecture (CORBA) to dictate the structure and operation of the radio. Critical components of the architecture are standards for loading waveforms, running applications and system integration. The SCA Hardware Framework (SCA/HW) dictates minimum specifications for hardware devices. These two frameworks assure software written to the SCA will work on compliant hardware. While this approach has some backing, there are numerous shortcomings. First, it has grown into a complex and heavy design. Second, it does not offer any capabilities for the emerging needs of a dynamic system such as a CR. Lastly, it is not well oriented toward lightweight implementations. The SCA is currently at version 3.0 and has been adopted as a standard by the Object Management Group (OMG). The current price tag on the development of the SCA has passed $1.3 billion and costs continue to increase. Additionally, the total production costs of individual components of the project are seeing drastic cost increases. The first to be delivered, the cluster 1 vehicle mounted radios, have risen from $15.1 to $21.6 billion [15]. The cost increases were attributed to daunting technological challenges.

2.1.3 Rapidly Deployable Radio Network

The Rapidly Deployable Radio Network (RDRN) was another early SDR (1994-1999) that was specifically designed to address problems when implementing mobile, rapidly deployable, and adaptive wireless communications [72]. Researchers created a high-speed ATM-based wireless communication system that was adaptive at both the link and network layers, allowing rapid deployment and response in a changing environment. Their concept of operations was to deploy a backbone of switches that would auto configure into an appropriate topology enabling users to access the backbone via a cellular-like architecture. The RDRN project incorporated digitally controlled antenna beams, programmable radios, adaptive link layer protocols, and mobile node management. Their network layer routing protocol when linked with the directional phase array antenna was the basis for their adaptive point-to-point topology solution. Additionally, at the datalink layer they were able to change modulation, frame lengths, and enable and disable forward error correction. However, project literature states that these parameters are set when the radio is initially configured. They concede that the next step for the project would be to develop an algorithm that would run in the radio processor, enabling dynamic adjustments to power level, coding depth, antenna pointing, modulation type, and data rate.

2.1.4 Virtual Radios/VANU Inc.

Around 1995 there was a large amount of interest in the transition from specialized hardware to general-purpose processors (GPP) in the radio space. In his thesis, Vanu Bose set out to demonstrate that it was possible to implement a high-data rate, computationally intensive, real-time signal processing application on a GPP. This effort formed the basis of his thesis and later launched his company, Vanu Inc [14]. His doctoral work produced a software-radio that was able to run on a general-purpose processor. In this work he acknowledges that there would be significant opportunities to improve wireless networking using software radio technology. He recognized that a radio which could adapt and change its operating characteristics in real-time could significantly impact wireless communication. In November of 2004, Vanu's software radio GSM base station became the first device to successfully complete the FCC's certification process governing software radio devices [91]. Mid Tex Cellular operates the first commercial deployment of this system. Vanu's Anywave GSM base station is capable of remote software updates, allows multiple standards to co-exist on one network, and has the potential to decrease cellular maintenance budgets by as much as 20% [97].

2.1.5 End to End Reconfiguratility (E2R)

The E2R project is one among a host of international efforts to realize the potential of C/SDR. The E2R project aspires to develop designs and prototype systems focusing on the end-to-end perspective. The project, now in its second phase, is focusing on producing platforms that are accessible across the device and user spectrum, ranging from advanced cellular handsets for the consumer to advanced spectrum management tools for the regulator [82].

2.1.6 SoftMAC and MultiMAC

SoftMAC is a software system developed at the University of Colorado by Neufeld et al [61]. This system was built to provide a flexible environment for experimenting with MAC protocols in the wireless domain. The ability to cheaply create, modify and conduct system level experimentation with hardware is often a goal of many research projects. However, many of these projects ultimately fail due to the cost, time, and effort involved in deploying a large scale experimental platform. The SoftMAC platform fills this need. It uses a commodity 802.11b/g/a networking card with a chipset manufactured by the Atheros Corporation to build a software radio with predefined physical layers but a flexible MAC layer. Internally, the Atheros chipset provides considerable flexibility over the format of the transmitted packets, though this flexibility is not generally exposed by network drivers. By reverse-engineering many of those controls, SoftMAC provides a driver that allows extensive control over the MAC layer while still allowing use of the waveforms defined by the underlying 802.11b/g/a physical layers.

MultiMAC, also developed at the University of Colorado, is intended to extend the basic SoftMAC environment to tackle problems in the areas of dynamic spectrum allocation and cognitive/software-defined radio [25]. It builds upon the functionality in the SoftMAC platform with some specific features in mind. First, MultiMAC allows multiple MAC layers to coexist in the network stack with minimal switching impact. Second, it allows one to dynamically reconfigure the MAC and physical layers on a per packet basis either from logic running as part of MultiMAC or from a user level process. Finally, by leveraging these capabilities MultiMAC allows intelligent reconfiguration of the MAC and physical layers, thus achieving a cognitive MAC. The cognitive MAC layer couples efficient reconfiguration afforded by MultiMAC with computational intelligence. This combination allows the engine to make smart decisions about which MAC layer should be used and which physical layer properties set.

2.1.7 Summary

This early research is important because of the progress made in the realization of the SDR. This research provides valuable insight into the design and implementation of future platforms. Additionally, the systems and research covered in this section give historical perspective and bolster the case for intelligent processing on a software radio. Much of this early work in SDR set the foundation for, and in some cases directly called for research to begin in creating intelligent radio systems. Generally, the research and development in software-defined radio has been characterized by the building of systems to solve specific problems. The DoD efforts are focused on waveform portability and radio interoperability. The RDRN was centered on mobile and rapidly deployable disaster response communication. Vanu Bose's work focused on solving problems associated with moving traditionally analog or custom ASIC components and processes to a general-purpose processor. Most of the problems in SDR lie in the development of more capable hardware and interoperable software frameworks. This base of adaptive protocol design has allowed for the development of the CU platforms, SoftMAC and MultMAC. Software radio research, although very active, has begun to give way to the recent popularity of cognitive radio.

2.2 Cognitive Radio

Scientific and commercial interest in cognitive radio stems from its "lofty" goals: to increase bandwidth, interoperability, and reliability through adaptive and efficient use of spectrum. In terms of spectrum policy (or the rules that govern spectrum usage), CR offers the ability to intelligently manage access to the spectrum. The popularity of CR in the research community is being fueled, in part, by this goal. Additionally, funding trends in DARPA and NSF have transitioned away from foundational SDR research to cognitive processes running on an SDR platform. It is easy to see that the funding trends and growth in interest are related, however, there have been few published works in the area. Early research has only just begun to tackle the issues raised when a system of these highly reconfigurable SDR platforms are acting in concert. Recall that the first commercially available software-radio platform was just recently made available in 2005. It follows that foundational work in C/SDR is often limited and focuses on the application of textbook techniques and simulations, rather than the genesis of entirely new algorithms.

2.2.1 Genesis Of Cognitive Radio

Joseph Mitola is credited with coining the phrase that is used to describe the joining of software-defined radio with computational intelligence, *cognitive radio*. He described how a cognitive radio could enhance the flexibility of personal wireless services through a new language called the Radio Knowledge Representation Language (RKRL) [39]. RKRL was a language for describing the features and capabilities of a radio. Mitola describes it as a set-theoretic ontology of radio knowledge. Mitola's doctoral dissertation was presented in May of 2000. His research resulted in development of an architecture for CR and formulation of a set of use cases for computationally intelligent radios. Additionally, he developed a simulation environment in order to test the viability of RKRL. The focus of the simulation was centered on natural language processing. The experience he gained while conducting this simulation lead to the formulation of his architecture for CR. This work set the stage for research in the area of CR by providing enumeration of what it means for a radio to incorporate computational intelligence. He described 9 levels of operation that relate to the functionality of a cognitive device [39].

- Level 0 - Pre-programmed
- Level 1 - Goal-driven
- Level 2 - Context Awareness
- Level 3 - Radio Aware
- Level 4 - Capable of Planning
- Level 5 - Conducts Negotiations
- Level 6 - Learns Fluents
- Level 7 - Adapts Plans
- Level 8 - Adapts Protocols

Indeed, DARPA's Next Generation (XG) project is now developing radio systems that operate at level 5 (conducts negotiations) and academics are working on devices that operate at level 7 (adapts plans). Mitola also described the possible steps that a CR might take in assessing its environment. These include observe, orient, plan, learn, decide and act. Various complex relationships might be defined among these steps. Mitola took a broad look at CR and the applicable algorithmic methods that could apply.

2.2.2 DARPA neXt Generation Communications

The Defense Advanced Research Projects Agency (DARPA) neXt Generation (XG) Communications Advanced Technology Office (ATO) is the project management arm of a DoD project whose objective is to develop and demonstrate a prototype spectrum agile radio [86]. They set a performance target of increasing spectrum utilization ten fold without causing harmful interference to non-cooperative radios. The project is addressing spectrum management difficulties associated with deployment and spectrum scarcity issues through development of devices and protocols for opportunistic spectrum access. In this scenario a primary user should not experience interference when a secondary user attempts to access a band. One method for mitigating inference by the secondary is to require the secondary to vacate the band and continually check for primary users. Much of the work in this area is dependent upon the development of highly sensitive detectors and the combination of individual and group sensing to determine the RF environment. Environmental data will be used to assist the device in making scheduling decisions about how best to operate. Other data could be derived from location, time of day, and databases containing pertinent data and policies on primary users. Web Ontology Language (OWL) will assist in the decision process by providing rules and behaviors in a machine-readable form. Together this information will allow the system to mark spectrum that is available for use.

A secondary objective of the program was to leverage and develop technologies that enable dynamic access to spectrum within constraints provided by machine-

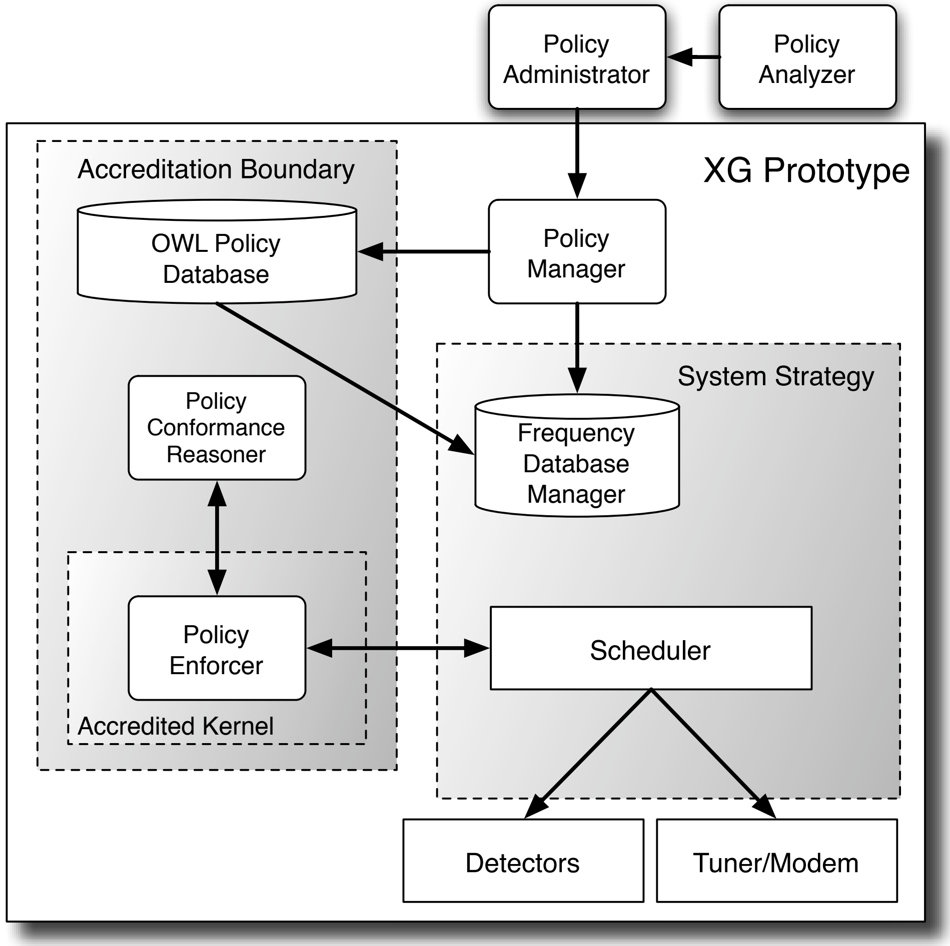

Figure 2.2: The DARPA XG Architecture

readable policies. Technology central to this goal includes adaptive MAC protocols, hardware independent policy-based reasoning and new waveforms. The policy engine operates by obtaining a set of policy conditions that might associate with a certain device, in a certain location, attempting to make use of a certain band. The policies will be expressed in extensible markup language (XML) and the engine will parse these policies to determine a limited set of possible operational conditions for the device. The device then determines how these might best be used to meet communication requirements. Each of these policies is authenticated and its operation traceable. Ultimately, the goal is to provide a framework that describes the policy boundary for a device, and allows the device flexibility to operate within these boundaries in a variety of ways. This system of policy driven operation is depicted in Figure 2.2.

Additionally, XG is pursing the creation of a waveform that combines non-contiguous narrowband channels. These waveforms will respond to and capitalize on spectral vacancies in both time and frequency. Higher in the protocol stack the system is able to adapt to future MAC layer protocols and algorithms. Here the difficulty lies in being able to maximize spectrum utilization while still allowing short duration changes in the frequency used without saturating the network with protocol overhead. The XG project's objectives are addressed through theoretical work, simulation and platform development.

2.2.3 Cognitive Radio Networks

Haykin provides a thorough overview of cognitive radio networks and describes the basic capabilities that a "smart" wireless device might offer [35]. Others describe techniques for applying CRs to improving the coordinated use of spectrum [10, 17]. Sahai et al., describes some of the physical layer limits and limitations of cognitive radios, including the difficulties associated with determining whether or not a radio frequency band is occupied [71]. Nishra has implemented a test bed for evaluating the physical and data link layers of such networks [62]. Additionally, Thomas describes the basic concept of a CR network and provides a case study to illustrate how such a network might operate [77]. It is also worth noting that the standards communities are focusing on cognitive radios. The IEEE 802.22 group is developing a wireless standard for the use of cognitive radios to utilize spectrum in geographically separated and vacant TV bands [23]. Also in the IEEE, the P.1900 workgroup is examining the general issue of spectrum management in next generation radio networks.

2.2.4 Algorithms for Cognitive Radio

This section details research in the area of applying intelligence to a software radio. The work ranges from somewhat simple techniques, including Listen Before Talk (LBT) and Dynamic Frequency Selection (DFS), to those more complex like game theory and genetic algorithms.

Often the word "algorithm" causes one to envision a complex and involved solution to a problem. However, there are many algorithms that are simple, elegant and applicable to C/SDR. Algorithms of this sort are members of a class of direct methods for tackling problems associated with the design and implementation of a C/SDR. Some of the more well know approaches are the LBT algorithms, all variants on the notion that the sender should sense (listen) to the media before attempting to access it [36, 43, 54]. Another technique, transmit power control, seeks to minimize the transmit power used among nodes decreasing overall interference. Dynamic Frequency Selection (DFS) algorithms have seen a corresponding rise in interest coincident with the increase in popularity of C/SDR. Horne, details four common approaches: (1) *Channel Availability Check* - before starting a transmission the sender must monitor the channel for a defined period of time in order to determine whether a signal is present (e.g., LBT). (2) *In-Service Monitoring* - here the radio device must continually listen on the channel by searching for signals between transmissions. (3) *Channel Abdication* - upon detection of a signal the device must stop transmitting and move to another channel. And finally, (4) *Channel Non-Occupancy* - if a channel is occupied, the sending device will not utilize the channel for a set time period [36]. These techniques are present in common everyday devices like many cordless 900Mhz phones. These phones sense interference on a channel and switch to another in order to obtain a clear signal.

Game theory relies on a mathematical model of an interactive decision process. Game theoretic research in C/SDR advocates an approach nearly opposite to that which is advocated in this work; rather than using simulation or experimentation to inform the development of the algorithm the game theorist uses the analytical power of game theory to guide their algorithmic decisions. In [58, 59, 60], Neel et al. describe their use of this approach. The research cited is concerned about bounding and qualifying potential algorithms according to a “game” that approximates the function of a cognitive radio. The authors contend that game models will give insight into algorithmic complexity. This approach to bounding potential algorithms in a defined game space is helpful; however, it does not appear to offer direct insight into what the underlying algorithm should be.

Other work in this area includes the application of machine learning or similar techniques to the problem of decision-making within cognitive radio networks. For example, Rieser describes a biologically inspired cognitive radio engine that employs genetic algorithms to optimize the robustness of the network [70]. He developed a cognitive model and architecture that was able to operate in unforeseen communication environments recalling past successful configurations via “memory”. Genetic algorithms are based upon the guided combination of “chromosomes”. In this case, a chromosome is a representation of a potential configuration of the radio system. His system would generate chromosomes and eventually settle on one that met the specified goal. System goals take the form of a weighted function, wherein numeric values are assigned to give importance to each variable. For example, to equally weight bit error rate and throughput one would give them the same value. The genetic algorithm was driven by a fitness function that was partially based on the weighted function. In genetic algorithms the fitness function is used to determine which chromosomes are passed on to successive generations of the algorithm, whereby the more fit chromosomes are combined in the hopes of forming more “fit” offspring at each successive generation.

2.2.5 Summary

Mitola's visionary work established a common framework for research in cognitive radio. Mitola took a broad look at cognitive radio and the applicable algorithmic methods that one could apply. He also developed a description language (RKRL) for the radio that establishes a common basis upon which one could layer a heterogeneous cognitive process. A prototype system was not built to test his architecture and theories. A Java simulation of one of his use cases was developed and incorporated RKRL. This simulation was focused on natural language processing in the C/SDR domain and is not applicable to this work [39].

A broad range of potential algorithms exists for controlling CRs. LBT and DFS algorithms are important because they are simple algorithmic approaches that afford huge returns. Nevertheless, they suffer from one-dimensionality in that they solely focus on interference free channel access. Investigation of simple algorithms like LBT across multiple dimensions could yield promising results. A collection of simple algorithms acting in concert could drastically outperform individual component solutions. The game theoretic approach is useful if one is interested in characterizing potential algorithms in the "game space", however, the algorithm behind this characterization remains a black box. The work presented here, informing the algorithm through experimental results, would only see minor benefit, if any, from a translation into the game theoretic space.

The results from Rieser's work in genetic algorithms indicates that a cognitive system can successfully reconfigure the radio in response to changing channel conditions, however, one critical piece of his work is not clear; the process of deriving the fitness function for his genetic algorithm. The fitness function used by Rieser was most likely based upon experimental results or was guided through informed design, thus lending support to the approach presented here. Beyond these described methods, numerous other techniques such as Markov models, neural nets, expert systems, and/or fuzzy logic could be applied. However, the question remains as to which of these will be successfully applied in a fielded system (this thesis attempts to answer this question in part).

2.3 Reliability Within a Network Protocol Layer

The intent of a communications network is to provide a means for the transport of data. A fundamental requirement for such transport is to provide a certain level of reliability - one that maps to the needs of the supported communication. Reliable communications can be broadly defined as the ordered timely reception of data without unacceptable loss. Timely, as it is used here, refers to the applications requirements on data delivery. For example, streaming media applications have much different timing needs than an application that is doing a bulk transfer of email messages. The reliability of a system may be compromised as a result of buffering problems within the network device, due to software/hardware errors, timing problems, corruption of bits, and/or interference.

Network reliability is commonly optimized through a variety of techniques at multiple levels of the protocol stack. The instantiation of reliability at the data link (i.e., HDLC) and transport layers (i.e., TCP - acknowledgments and resends) is common, but can also be implemented at the physical and network layers. Physical aspects of a network environment have significant impact on reliability. Attributes of the physical layer can be optimized to ensure low bit error rates. To counteract these differences, wireless protocol stacks rely on signal processing, media access control, and routing to improve overall reliability. When used in combination these techniques can improve the reliability of the network significantly. However, in practice, these measures are designed to operate within a layer and do so without any understanding of the environment or knowledge of the other layer's capabilities.

Signal reliability can be improved through adjustments to power, diversity and direction. Other techniques, including bit rate adaptation and custom signal processing, can also be used to improve the reliability of a signal. At the data link layer, error detection, correction and flow control can all be used to improve reliability. At the networking layer, routing mechanisms can be applied to ensure reliable route connections - for example, flooding techniques can be applied to force multiple copies of a message across different network paths, with the hope that at least one will be successfully received. At the transport layer, there are numerous methods for improving reliability including, checksum techniques, timers, sequencing numbers, acknowledgments (both positive and negative) and windows. This following subsections provide a high level survey of reliability enhancing techniques with respect to the network protocol stack.

2.3.1 Physical Layer

Reliability of a network depends heavily on the type and characteristics of the physical media. Characteristics of optical networks are substantially different from those of wireless networks. Specifically, optical networks have bit error rates (BER) approaching 10^{-12} whereas radio networks can commonly experience bit error rates of 10^{-4}. In wireless networks, bit errors (and as a consequence packet loss) may be caused by attenuation, inter-symbol interference, doppler shift and multipath fading [5]. Furthermore, in wireless environments it is common to see bursty disturbances as opposed to the stochastic interference characteristics of wired systems. Radio networks must also be able to adjust to physical phenomena such as reflection, diffraction and scattering.

Various techniques can be applied to the physical layer to improve reliable transport including power adjustment, changes in directionality, and encoding techniques. Several researchers have considered the role of directionality as a means of improving the performance of wireless systems [46, 100]. Moving from omni-directional to directional transmission allows for substantial spatial diversity improvements [66]. One technique for improving successful reception of a signal over distance is to increase signal power. However, increasing the signal comes with a price. You might (1) overwhelm the receiver or neighboring nodes, (2) use more signal than is necessary, (3) violate the law (federal spectrum laws) and/or (4) consume unnecessary power (e.g., battery drain).

2.3.2 Data Link Layer

Various reliability mechanisms exist at the data link layer including link layer protocols, flow control, and error detection and correction. Early work in the area sought to design link layers for generally stable networks [38]. While in [3], Awerbuch et al., described algorithms to adapt to expressly unreliable links. A variety of link layer protocols can assist in improving the reliability of communications including, Automatic Repeat reQuest (ARQ), Channel Partitioning Protocols, Random Access Protocols and Turn Taking Protocols [50]. Each of these has their strengths and weaknesses. Certain types of reliable link layer protocols offer flow control. The basic idea is to throttle the sender until the receiver is ready to accept more frames. Both Stop-and-Wait and Sliding Window techniques can be used to prevent the sender from transmitting until the receiver provides permission [50].

Coding techniques can be used to improve the reliability of a received signal and to detect and/or correct errors. Error detection is commonly thought of in terms of parity checks, Cyclic Redundancy Checks (CRC) and other check-summing methods. Lin and Costello, provide a broad overview of basic error control coding techniques [50]. Techniques for one-way error correction are generally based on Forward Error Correction (FEC), which provides error-correcting codes that can automatically correct an error at the receiver. FEC includes such techniques as linear block codes [34], cyclic codes [65], BCH codes [69] and convolution codes [27]. These coding schemes differ substantially in their approach and application. Various researchers have focused on encoding in the wireless space. Work by Zorzi and Rau looks at probing ARQ techniques to lower retransmission in fading environments [101]. Other work by Ayanoglu et al. considered the asymmetric design of ARQ protocols where the complexity resides in the base station [4]. It is well understood that link layer error recovery techniques are required to provide reliable data flows over wireless links. The reliance on transport layer recovery techniques results in poor throughput, loading problems, and inefficient use of the wireless link [26, 51, 52].

The length of the frame can have an impact on reliability. For example, a very large frame might be more efficient for large blocks of data, but if the frame is corrupted and a retransmission is required, the cost in overall channel efficiency can be significant. Various researchers have explored adaptive frame length as a mechanism for improving throughput and range [26, 49]. This work shows that adaptive frame length control can be exploited to improve throughput in the presence of noise.

2.3.3 Network Layer

The network layer allows packets to be routed through a network or internetwork to the proper destination. The goal of a routing protocol is to ensure that devices know what nodes to direct packets through to a destination. Huitema, provides a detailed survey of IP routing [37]. While most routing is based on the transmission of a packet along a single route to a destination, other techniques allow a packet to be replicated and flooded across numerous routes to ensure delivery. In [67], Ramanathan et al. provide an early survey of routing techniques for mobile environments. More recent work by Johnson [42] describes an early effort at mobile routing protocols for IP based networks. Also, in [66], Ramanathan examines the performance of ad hoc routing protocols for beam-forming antenna networks.

2.3.4 Transport Layer

In [30], Garlick et al. describe general problems and techniques for improving reliability at the transport layer. They discuss issues of flow control and sequencing in detail and provide general techniques for addressing these issues. A more recent treatment of the subject includes discussion of adaptive bandwidth control and discussion of TCP in wireless environments [6, 16, 74].

A number of techniques exist at the transport layer to ensure reliability, including checksums, timers, sequencing numbers, ACK and windows. Checksums are often used at both the data link and transport layers. Checksums can include a broad range of methods from simple parity checks to complex cyclic redundancy checks [47]. Timers allow for a threshold to be set after which the assumption is that the transmission or acknowledgment failed. Sequence numbers allow for the accurate ordering of transmitted packets. In this scheme buffers hold packets while making requests for retransmission of erroneous or failed packets. Acknowledgements serve to inform the sender whether a packet has been received or a retransmit is required. Windowing allows for a negotiated increase or decrease in the number of packets sent over a period of time. This allows a receiver to throttle the sender while it processes packets [47].

Transport protocols such as TCP, while intended to be independent of the underlying network stack, were tailored to the operation of wired systems. Wireless networks differ from wired networks in significant ways. First, in wired networks lost Protocol Data Units (PDU) are typically a result of router discard due to congestion, whereas in wireless networks lost PDUs are typically a result of bit errors or handoff failures. Therefore, rather than apply congestion control (to minimize congestion at the router), a better technique is to detect the loss quickly and retransmit [7]. In addressing this difference, two solutions have emerged. One approaches the problem by suppressing congestion control [53] and the other tries to hide the problem by presenting a lower effective bandwidth (throttling at the link layer) [8]. A second difference between wired and wireless networks are the typical power and computational constraints. This has motivated the design of lightweight transport algorithms such as Mobile-TCP [33]. A more challenging consideration is to provide a transport protocol that operates well across both wired and wireless links, which represents an increasing percentage of network traffic. This type of protocol will require algorithms that can adapt to changes in the network links as the transmission occurs, much as the work proposed here.

2.3.5 Summary

It is apparent from this survey of reliability techniques that there are a host of potential avenues to explore and consider in development of an experimental platform and algorithmic solution. The various techniques cited above were all considered in the design of cognitive engine presented as part of this work. The major difference between the work referenced here, and the solution presented, is the extension of reliability from an intra-layer process to one which spans layers. C/SDR provides a mechanism for exploring the effects of cross-layer reliability techniques. For example, you could have a directional antenna change routes to move traffic around a pocket of interference while turning on FEC and changing framesize. The experimental approach presented provides valuable insight into the potential of C/SDR to realize beneficial cross-layer effects.

2.4 Dynamic Protocol Adaptation

This section describes efforts within the research community to improve system performance through cross-layer adaption. This section also describes work on hybrid and flexible MAC design.

2.4.1 Cross-layer Optimization

Research in the area of cross-layer optimization in wireless systems has been an area of focus in recent years. Others have also spent a considerable amount time and effort investigating cognitive radios. However, the potential of improving the performance of a wireless system by combining cross-layer optimization with cognitive systems is just emerging as a research area. Since its inception, the layering principle, which segments network functions and specifies standard interfaces between layers, has served the networking community well.

There is a long research history of optimizing such functions within a layer, in a sense incremental-intra-layer optimization. This work includes such efforts as improved encoding techniques at the physical layer, novel media access techniques at the data link layer, adaptive routing design at the network layer, and enhancements to retransmit algorithms at the transport layer. However, various challenges, such as demand for bandwidth, the unsuitability of the standard TCP/IP protocol suite for wireless networks, the Internet commerce boom, etc. are driving researchers to look more closely at how the traditional networking layers interact and function. Much of the networking protocol research has been wire centric. The advent of wireless systems, with fundamentally different physical and data link layer characteristics, is pushing the need to rethink boundaries and the typical isolation of layers in traditional network protocol design. Work in the area of cross-layer optimization has typically focused on enhancing throughput, Quality of Service (QoS) and energy consumption [9, 31, 41]. However, these types of cross-layer optimizations tend to be limited in their application. In fact, most of the work in this area tends to focus narrowly on two layers, producing unique solutions. In other words, they do not incorporate a broadly adaptive solution nor do they consider user level requirements.

Vadde et al. have applied response surface methodology and Design of Experiments (DOE) techniques to determine the factors that impact the performance of mobile ad hoc networks (MANETs) [78, 79, 80]. Their research considers routing protocols, QoS architectures, media access control (MAC) protocols, mobility models and offered load as input factors and throughput and latency as response factors. Their analysis demonstrates the usefulness of these techniques and shows where certain input factors can outperform others within a MANET.

Kawadia and Kumar present an interesting critique of cross-layer design in [45]. They warn that cross-layer optimization presents both advantages and dangers. The dangers they discuss include the potential for spaghetti design (in other words, complex seemingly unstructured design), proliferation problems and dependency issues.

2.4.2 Flexible and Adaptive MAC Design

MAC is a sublayer of the data link layer and the logical interface between the physical and link layers. The MAC is concerned with such things as link level addressing, frame limiting, error detection and control of the physical transmission [75]. Many of the more popular MAC protocols are defined by the IEEE 802 group [84]. In very simple terms, the MAC layer frames data and passes it to the physical layer. The task of arbitrating access to the physical medium is carried out through a variety of techniques embedded within the MAC layer. Commonly known algorithms for controlling access to the physical layer include Aloha, DQDB, CSMA/CD, CSMA/CA, and Token-ring. Some of these techniques are fairly simple, such as Aloha, wherein when frame arrives the MAC immediately forwards it to the physical layer for transmission. Other MACs like CSMA sense if the media is busy before transmitting.

802.11 defines a set of MAC specifications for wireless LANs. It provides coordination of access among clients and the base station by specifying a set of expected operations for access and reservation of the link. Primary 802.11 MAC functions include: scanning for channels, authentication and privacy, link control, access point association, RTS/CTS reservations, power control and packet fragmentation [1]. To gain access to the media, the standard defines two methods - distributed coordination function (DCF) and point coordination function (PCF). DCF is based on CSMA/CA and relies on a reservation timer (referred to as the Network Allocation Vector or NAV) to indicate when a transmitting client will be done. Clients must determine the time it will take to complete transmission and include this time in the duration header of the frame. DCF also includes a random backoff timer to help prevent waiting clients from simultaneously starting transmission after the channel becomes available. Acknowledgments from the receiver allow the sender to know whether or not there was a collision. PCF on the other hand provides a polling technique, wherein the access point polls clients for their access requirements and assigns usage (providing support for synchronous data like video) [1].

Work on enhancements to wireless MACs is described in various papers. This prior research has shown that network efficiency can be increased by using multiple MAC layers simultaneously. Other work has shown that problems inherent in 802.11 networks can be improved upon by "overlaying" one MAC protocol on top of an existing protocol [68]. Finally, a variety of hybrid techniques and subtle variations on existing MACs have been proposed [28, 29, 56, 73, 32, 99, 57, 11, 12, 44, 55, 63].

Various hybrid MAC approaches are described in [73, 32, 99, 57, 11, 12, 44, 55, 63]. These approaches combine CSMA like bandwidth requests with scheduled allocation of bandwidth. That is, they incorporate the aspects of two separate types of MACs to form a composite MAC. Similar to these hybrid approaches there has also been work done in creating an overlay MAC layer (OML), which attempts to overcome the fairness and allocation problems inherent in the 802.11 MAC [68]. In [28, 29, 56], Farago et al. describe MetaMAC, a method for automatically selecting an optimal MAC protocol among a set of existing protocols. Their approach is to create an adaptive layer that sits on top of a set of existing MAC protocols. Rather than relying on a centralized controller to coordinate the selection process, MetaMAC makes use of a local feedback mechanism in selecting which MAC to apply to a packet. MultiMAC, described later, borrows from and extends this MetaMAC concept. An important initial difference between this work and MultiMAC is that MultiMAC is an actual implementation of an adaptive MAC layer. MultiMAC also provides flexibility in how and when the MACs are selected. For example, selection of a MAC could be driven by changes at the MAC layer (e.g., frame errors) or a user level process making requests for reliable transmission.

2.4.3 Summary

Hybrid MAC protocols are typically one-of-a-kind solutions. That is, a new "MAC" is created that embodies a set of requirements in order to attain a desired goal. Solutions of this nature must all interoperate at the MAC layer in order to function.

As described later, the experimental platform used in this thesis exports an interface to the user level where the rules for deciding which MAC to use for transmitting a certain packet are viewable and editable. Unlike the MetaMAC and overlay MAC approaches, neither the number and selection of MACs nor the rules to choose between them are "hard-wired" into the system. This allows one to start with a set of criteria for the selection of a MAC layer. Depending on the context, one could also adapt and modify criteria using a case-based reasoning algorithm, thus transitioning to a *cognitive platform*.

MetaMAC attempted to optimize the ability of a specific MAC protocol to access the media, only considering the success or failure of the access. This limits the range of possible MAC behaviors. For example, a MAC protocol that provides forward error correction with CSMA may be less successful at acquiring the media than a protocol with no error correction that uses oblivious access. On the other hand, in a noisy environment, the protocol with error correction would actually accomplish more. The complication in this example is that stations can only estimate the "useful work" or "yield" of packets they receive, unless the channel conditions are symmetric, some additional protocol may be needed to communicate this end-to-end information.

2.5 Focus of Related Work

There is a large amount of prior work and active research that can be brought to bear on the problem of developing a system that is able to improve wireless performance in the presence of a jamming node. This work is particularly focused on research in the areas of cross-layer adaptation, interlayer adaptation, and methods for tuning the C/SDR. Key concepts that were applied to this research are the use of adaptive mechanisms (data rate, power), cross-layer exploits (selective queueing), and tuning of the cognitive algorithm (informed case-based reasoning). The methods that provided the most impact were targeted for use in this thesis. The next chapter will highlight their use.

Chapter 3

Method

Cognitive radios offer a broad range of opportunities for improving the use and utilization of radio frequency spectrum. Affording the opportunity to create radio networks that can reconfigure their operation based on application requirements (e.g., latency and/or throughput), environmental conditions (e.g., noise floor) and/or operational policies (e.g., commands to vacate a particular frequency band). Such reconfiguration requires an understanding of cross-layer interactions within the network protocol stack. These devices will also require the development of algorithms to determine when to reconfigure and investigation of how this change impacts the network. In developing such algorithms, it is necessary to understand the implications of varying parameters at the physical, data link and network layers. While it might seem intuitive to increase the transmit power of a radio to ensure that it is heard by the intended recipient, this increase could also harm the communication of other nodes in the area. Furthermore, it might not be beneficial to enable a forward error correction on a channel with low bit error rates. Thus, it is necessary to understand the implications of **turning the knobs** on the radio prior to doing so. The method used in this work is founded on this principle.

The research conducted in this thesis follows a three phased approach. The first phase consisted of decomposing the larger research problem into smaller more manageable pieces. The second phase, the experimental work, was composed of the

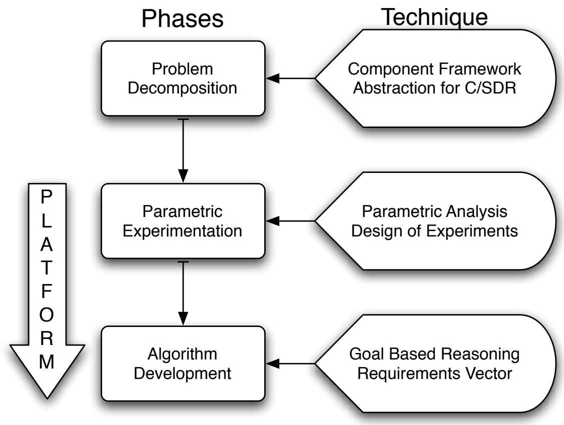

Figure 3.1: Research Method

data collection and analysis. In this phase the parameters of the C/SDR are varied and their impact on the metrics of interest is observed. The third phase builds upon the work done in phase two with the design, development, implementation, and evaluation of an algorithm for controlling a C/SDR. A task which runs parallel to the final two phases, is the development of an experimental platform upon which the parametric runs and algorithm development and evaluation could be performed. The remainder of this chapter includes a discussion of the assumptions made in conducting this research, the problem decomposition, the parametric experimental work, the cognitive algorithm, and the experimental platform. Figure 3.1 provides an overview of the process.

3.1 Assumptions

The assumptions detailed in this section are largely based upon approximating the function of 802.11 and a typical military scenario in which a secure base of operations is communicating to a forward location. The following is the set of assumptions and limiting factors that served to focus the research:(1) The system will only deal with a single hop wireless connection. There is a host of multi-hop routing, MAC, and other research supporting the consideration of a multi-hop implementation; however, in order to maintain consistency with a standard 802.11 device, single-hop was chosen. Single-hop is in-line with current 802.11 operation, as both station to station and station to base station communication are single-hop. (2) Communication nodes will keep their position static. Similar to single hop this limitation reflects the way most 802.11 users operate. WiFi users tend to remain in one position for minutes rather than moving constantly. (3) Experimentation is limited to node-to-node communication. (4) The work is focused on what a C/SDR can accomplish to maximize its performance within a frequency band. (5) Finally, the layout of the nodes is such that the effects of the jammer node are primarily localized at one of the two communicating nodes (e.g., in this work the jammer effects are felt primarily by the client node). This was done in an effort to mimic a typical military communication scenario. In this scenario, a relatively secure server node (e.g., a military base) is attempting to communicate with a position that would be closer to opposition forces, therefore subject to a jamming source (e.g., a forward location).

3.2 Problem Decomposition

This section presents the initial phase of the research method. This phase is primarily a thought experiment, therefore the conclusions serve to provide background for the successive phases in the approach. This section begins with a presentation of the component architecture, which was developed to further understanding of how one could decompose cognitive functionality in a system of radios. This is followed by a presentation of a method for representing application requirements and/or limitations. This representation forms the basis of a method that could be used by a C/SDR in conjunction with parametric knowledge to determine desirable configurations.

3.2.1 Component Framework

This framework is based upon decomposition of the requirements and tasks that a cognitive software-defined radio should be able to perform. As with any framework, the goal is to create a manageable way of conceptualizing the object of study. A C/SDR incorporates a myriad of inputs and outputs and their interdependencies, similar to any large software system. The framework divides the problem in a way that makes these interactions manageable, while not restricting the potential flexibility of the system. Additionally, the framework addresses problems that arise in a system which is composed of radios with differing capabilities and requirements. The guiding principle in developing this framework was that the C/SDR needs to compute a *desirable* configuration based on policy, the environment, requirements, and the radio's capabilities. It is also beneficial to theorize on how a fully functional C/SDR might operate in order to provide a basis from which to focus on specific research questions. Figure 3.2 is a graphical depiction of the component framework. A description of the individual components of the framework follows.

- **Requirements Collection Component** - User and application needs will be a large factor in computing a *desirable* configuration. This component is responsible for collecting these requirements and getting them to the component that will make decisions about reconfiguration. One can imagine requirements such as, bandwidth needs, jitter tolerance, latency, pay for service parameters, security classification, and priority.

- **Policy Component** - One of the promises of C/SDR is the ability to dynamically incorporate policy. This component is responsible for ensuring that the system can react to changes in legal operating parameters. These parameters may change based upon geography, emergency conditions, and/or changes in law.

- **Capabilities Component** - This subsystem is responsible for collecting the capabilities of each of the C/SDRs in the network. This component would collect a description of the hardware and software characteristics of the C/SDR.

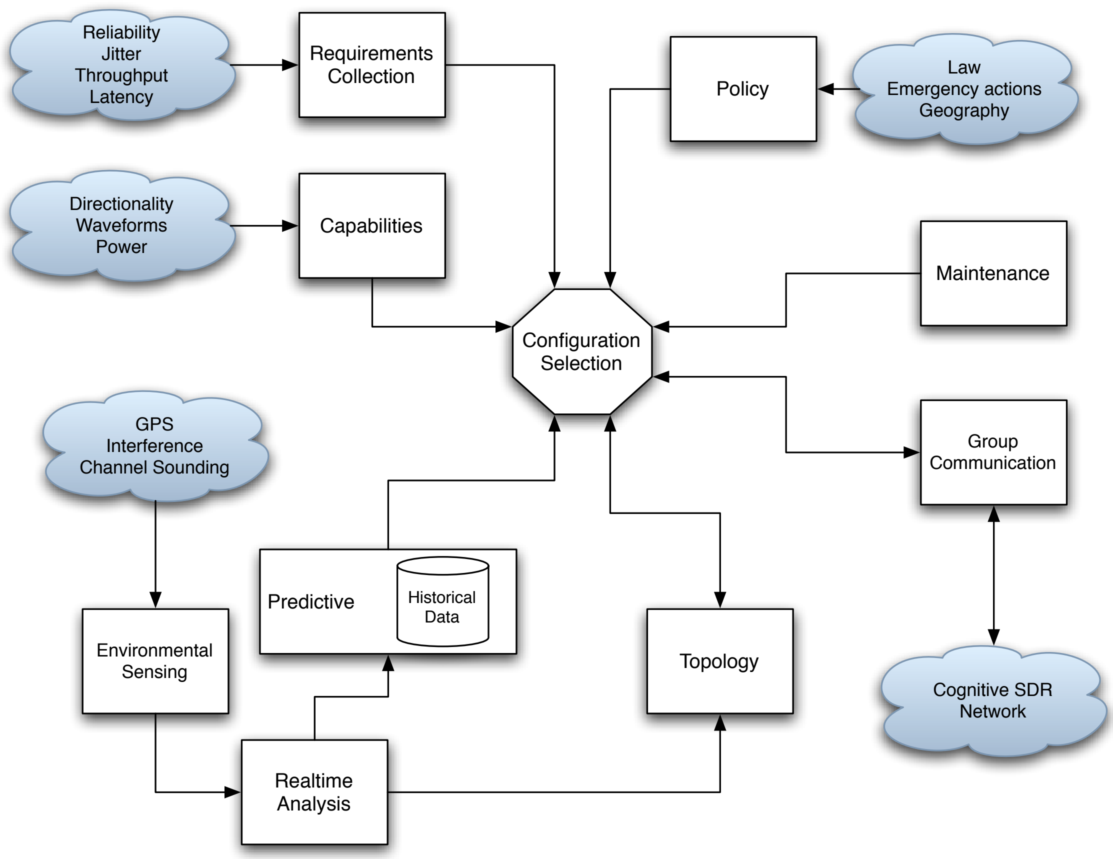

Figure 3.2: Component Framework for Cognitive Software Radio

A language like Mitola's Radio Knowledge Representation Language (RKRL) could be used to specify the radios capabilities [39].

- **Real-time Analysis Component** - This subsystem would monitor real-time performance of the C/SDR network in order to effect configuration changes. Collected data would feed into a predictive component.

- **Predictive Component** - This subsystem analyzes past configurations and data to reason about proactively reconfiguring the C/SDR network. For example, one common problem in metropolitan areas is cell saturation during morning and evening commute times. Using this component's functionality the C/SDR could proactively act during these times to alleviate congestion.

- **Environmental Sensing Component** - This system is responsible for collecting information about the C/SDRs environment. Technologies like GPS and/or beam steering could be used.

- **Topology Generation Component** - This component will generate topologies based upon input from the other components. Directional antennas, changes in power output, and dynamic routing will vastly improve the efficiencies of the C/SDR through spatial reuse.

- **Configuration Selection Component** - This system determines the best configuration of the C/SDR and informs the nodes in the C/SDR network, via the Group Communication Component, of the reconfiguration.

- **Maintenance Component** - This component is responsible for managing the software running on the C/SDR. This would include installation of software upgrades and patches or other maintenance activities. Ideally, this could be accomplished remotely through over-the-air/over-the-network updates.

- **Group Communication Component** - This component is responsible for communicating across the C/SDR network.

A C/SDR system may or may not incorporate all of the functionality detailed in this framework. The research presented in this thesis incorporates aspects of each component with the exception of the topology, policy and maintenance components.

3.2.2 An Abstraction for C/SDR

One of the research questions that must be answered when a C/SDR reconfigures, is how should one specify the requirements and limitations of the application to the cognitive process. How well the network is able to meet the needs of the application is dependent on the radios capabilities. The ability to dynamically redefine the lower layers of a radio device offers tremendous opportunity to improve reliability [22]. This is in stark contrast to the static nature of traditional radio devices, which tend toward fixed operational modes and a potential for less efficient use of the available RF spectrum. Beyond the technical and regulatory limitations, the static nature of the protocol stacks associated with these devices further limit their potential efficiency. This type of inefficiency is often due to higher layers making incorrect assumptions about lower layers and channel conditions. Such inefficiencies are further exposed when the protocols are evaluated against new metrics such as energy efficiency, overhead or impact on the noise floor. As a result, and as described in Section 2.4.1, cross-layer approaches to overcome these deficiencies have become a common theme in the literature [31, 9, 41, 45]. A C/SDR is one vehicle for capitalizing on these cross-layer interactions.

Such cross-layer interactions occur at different layers of the network protocol stack. For example, TCP may depend on the link layer for information about the cause of packet loss or expiration of timers. In the absence of such knowledge, TCP may relate the cause to network congestion. In reality it might be that transient noise introduced extra errors. Similarly, one may depend on the routing, link and physical layer to provide QoS. The routing layer may try to use multiple routes while the link layer may assist by choosing less congested links. Originally, many protocols were designed with little consideration of the properties of lower layer layers of the protocol stack, for example, application protocols viewed wireless networks as being similar to wired networks. However, the link and physical layers play a significant role in achieving good performance in wireless networks. For example, choosing a higher capacity link at the physical layer or avoiding nodes with high link-layer contention can improve throughput dramatically [9, 41, 45]. Other desirable network performance metrics may also be met through cross-layer interactions. For example, energy consumption, though a physical layer property, may depend on the needs of the higher layers. A routing protocol may vary transmission power depending on its need to reach just one or many nodes [31, 45].

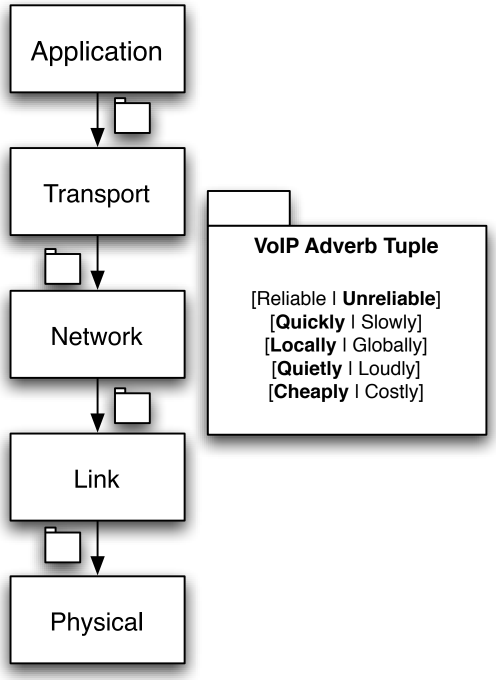

Figure 3.3: The VoIP Adverb Tuple is Accessed Across the Layers of the Protocol Stack to Affect Changes in the C/SDR.

One may ask how should such cross-layer interaction be expressed. In other words how should the requirements, goals, and limitations of the application or user be communicated to the C/SDR. One could abstract the higher layer interaction from the lower layers using *adverbs*. In the traditional linguistic context, adverbs are used to modify verbs, adjectives or other adverbs. In the proposed model, one applies the adverb analogy to modifying verbs associated with communication. For example, one might want the data to be sent **quickly**, **reliably**, or **locally**. Similarly, the properties of the layers can be abstracted using an *adjective*. Again, in the traditional context adjectives are used to describe nouns. In our model, an adjective is used to describe a communications attribute or goal. For example, a network link can be **capacious**, the medium can be described as **noisy**, or a requirement like **reliable** placed on the C/SDR. Fig. 3.3 depicts an adverb **tuple** that could be used to influence the cognitive process in the software radio. Additional details on adverbs and adjectives can be found in the proceedings of the 2005 International Symposium on Advanced Radio Technologies (ISART) [93]. The section which follows describes the experimental design, data collection and analysis portions of the method.

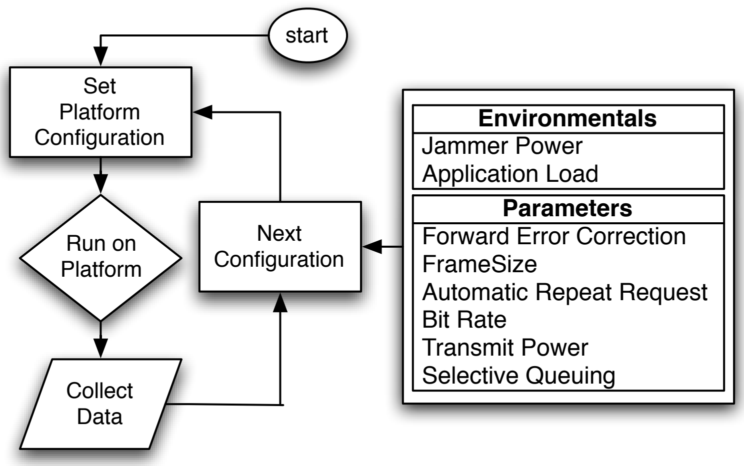

Figure 3.4: Parameters and Environments are Changed for Each Run

3.3 Experimentation

Phase two of the method requires the characterization of how the C/SDR's settings affect its performance. This portion of the research followed the basic procedure shown in Figure 3.4. The procedure is as follows: (1) One of the possible permutations of the radio/environment is selected. This configuration includes environmental factors (i.e., application load, jammer power) as well as those that are internal to the C/SDR (i.e., forward error correction, bit rate). (2) This configuration's performance is evaluated on the experimental platform and data collected (i.e., how did this setting affect throughput). (3) This process is repeated until every possible configuration of the system is examined. This basic experimental procedure is followed by analysis of the data using simple averages. The average effect of a parameter is calculated across all possible configurations. For example, this technique allows for determining the effects of enabling or disabling forward error correction on throughput. This simple technique, though useful, should be built upon by utilizing Design of Experiments (DOE) and Analysis of Variance (ANOVA) methods [2]. The subsections which follow discuss the node layout used for the experimentation, the variables involved, the set of C/SDR parameters, the evaluation metrics, and the techniques used to analyze the results.

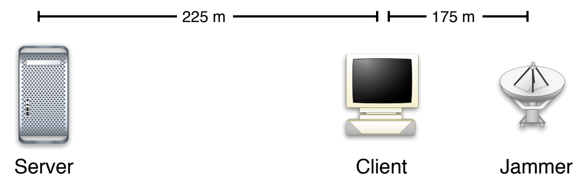

Figure 3.5: Experimental Layout

3.3.1 Layout and Setup

The experiments consisted of two nodes communicating in the presence of a noise source (e.g., a noncooperative node on a different network, an environmental noise source or a jammer). Figure 3.5 shows the physical layout of the two communicating nodes in relationship to the noise source. The distances chosen were selected to isolate the server node as much as possible from the effects of the jamming node. The physical layout of the nodes and noise source is fixed across all of the runs. As mentioned earlier, the potential configurations of the experimental system are dependent upon two sets of variables, those parameters and settings which are external to the C/SDR and those which are internal. Additionally, there are a set of metrics that are used to evaluate the performance of the system.

3.3.2 External Parameters and Environmental Variables

This section details those parameters and variables which were external to the C/SDR in our experimental work. Table 3.1 provides a list of those parameters and their settings.

Table 3.1: External Parameters and Environmental Variables

Parameter	Settings
Jammer Noise Level	1,4,11 mW
Offered Load	0.5,1.5,5.5 Mbps

- **Jammer Noise Level** - The uncooperative (or jamming) node is emitting noise bursts in a Poisson distribution centered around an interarrival time of 0.0125 seconds and a burst length of 2048 bits at one of three power levels. In the development of the experimental platform, a broad range of noise parameters including different distributions, inter-arrival times and power levels was analyzed. A group of settings that provided appreciable interference without overwhelming the communicating nodes was used.

- **Offered Load** - Load on the system was generated from two applications running on the server node. The first application was a constant bit rate source designed to mimic the transmission of a file from the server to the client. The second application generated a stream of data designed to mimic a Voice over IP (VoIP) load on the system (with the call originating at the server and terminating at the client). The requirements and needs of these applications were distinct, in that VoIP is very inelastic with respect to latency and jitter, whereas file transfer is tolerant of large variances.

3.3.3 Internal Parameters

This section details those parameters and variables which were internal to the C/SDR. Table 3.2 provides a list of those parameters and their settings and the respective layer in the protocol stack.

- **Selective Queueing** - This parameter affects the way in which packets are treated in the transmit queue. If enabled and the system recognizes that the

Table 3.2: Internal Parameters

Factor	Levels	Layer
Selective Queueing	off/on	Network
Automatic Repeat Request (ARQ)	off/on	MAC
Framesize	2048,9216,18432 bits	MAC
Forward Error Correction	off/on	MAC
Bit Rate	1,2,5.5,11 Mbs	Physical
Transmit Power	5,32,100 mW	Physical

packet is a VoIP packet it will enqueue the packet at the front of the transmit queue, thereby giving preference to latency sensitive applications. Otherwise, the packet is enqueued at the end of the transmit queue.

- **Automatic Repeat Request (ARQ)** - ARQ causes the receiver's media access control layer to send an acknowledgment frame confirming that it has received data from the sender. If the acknowledgment is not received the sender will retransmit the frame. In our experimental platform the system will retry a frame several times before failing. Failure is not reported and the system transmits the next frame in the transmit queue.

- **Frame Size** - This setting determines the maximum size of a transmitted frame (at the media access control layer). If the MAC receives a frame that is larger than the current setting, the frame is fragmented by the MAC based on the current max frame size. For example, if the max frame size is set at 2048, and an application sends a 4500 bit frame, the MAC layer would fragment the application data into two 2048 bit frames and one 404 bit frame.

- **Forward Error Correction** - When enabled this parameter uses a forward error correction scheme to encode the data. A forward error correction scheme adds parity overhead to the data in order to enable bit-error recovery at the receiver. A Reed Soloman code was modeled in the experimental platform [50].

- **Bit Rate** - This is the transmit rate of the system, measured in bits per second. There are four values used for bit rate. These values were chosen to mimic those available to an 802.11b wireless physical layer. As in 802.11, the experimental platform also adjusts the modulation accordingly as bit rate changes.

- **Transmit Power** - This is the transmit power of the transmitter measured in milliwatts.

There are a significant number of parameters and their settings. This set of parameters and environmental variables requires 2,592 experimental runs to provide complete coverage of the C/SDR's configuration space. The next section details the performance metrics used to evaluate the experimental platform.

3.3.4 Metrics

The performance data used to evaluate the system was collected from the perspective of an application making use of the platform. Three sets of statistics were collected, an overall measure, a measure from the perspective of the file transfer application, and a measure from the perspective of the VoIP application. For example, if the system's total throughput were measured at 200,000 bps this could be composed of a 90,000 bps VoIP stream and a 110,000 bps file transfer stream. Overhead at the MAC layer is encompassed in the statistics reported. For example, if ARQ was enabled this could potentially result in a lower bit loss at the application layer as well as higher latency (due to the acknowledgment frame exchange). Each of the metrics is shown in Table 3.3.

- **Bit Loss** - This is a measure of how many bits were lost during the run. It is reported as a percent.

- **Latency** - This is a measure of the average length of time for a packet to reach its destination during the run. It is reported in seconds.

Table 3.3: The Set of Metrics

Metric	Units
Bit Loss	Percent
Latency	Seconds
Jitter	Seconds
Throughput	bps

- **Jitter** - This is a measure of the variance in latency from frame to frame. The average for the experimental run is reported in seconds.
- **Throughput** - This is a measure of the number of bits transmitted successfully over the duration of the experimental run. This is measure in bits per second. When taken from the perspective of the application layer, as reported in this thesis, this measure is known as goodput, as all overhead is taken out of the measure (i.e., protocol bit overhead, retransmits, etc).

The following sections detail the methods used in analyzing the experimental data.

3.3.5 Analysis with Average Effect of a Parameter

The initial approach to looking at the data collected during the experimental phase was to use a simple average across all runs to determine the effect of a parameter setting. A chart from chapter 4 is provided as an example in Figure 3.6. This chart shows the average effect of changing parameter settings on VoIP latency. The effect of enabling Selective Queueing on VoIP latency is quite pronounced, the average latency across the runs when selective queueing is disabled is over 22 seconds compared to nearly zero when it is enabled. Latency is very high in this example due to overloading of the system and queueing at the MAC layer. This technique is a good way to get a sense of the effect of changing a parameter on the system. This method has some pronounced weaknesses in that it does not identify interactions between parameters, nor is does it

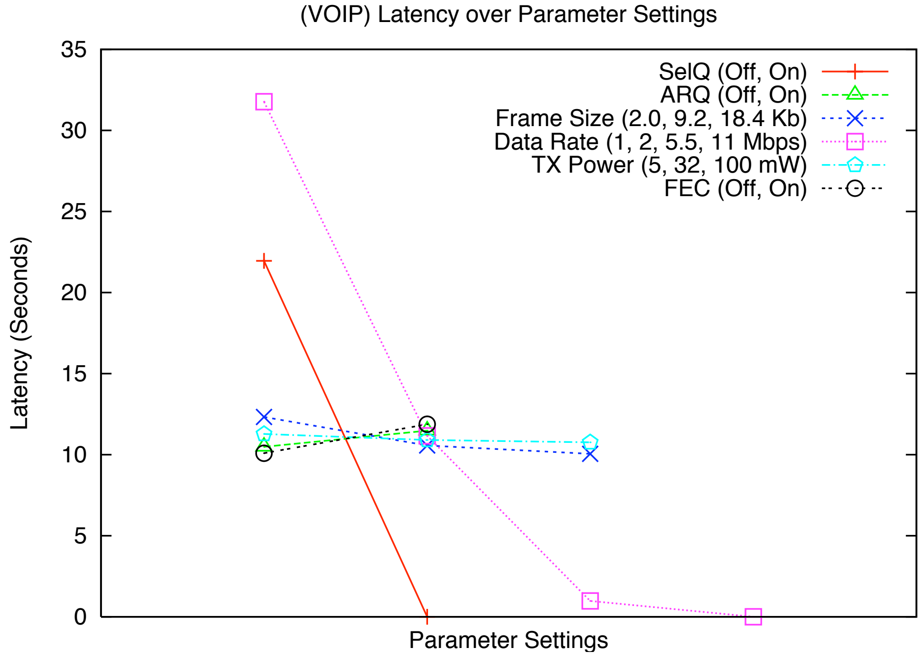

Figure 3.6: Average Effect on VoIP Latency

have the statistical power required to produce an accurate predictor of the system's performance. For these reasons DOE and ANOVA methods were also used.

3.3.6 Analysis with Design of Experiments

One should be able to predict or determine which parameter settings are able to deliver a specified performance goal based on a set of possible system configuations, environmental conditions and an expressed demand. In practice, this would be a continuous process, as illustrated in Figure 3.7, or alternatively it could be used once during a pre-deployment training phase. Such a process requires the development of a predictive model to configure the C/SDR. The radio is then used for communication and data on the achieved performance goals is recorded. The collected performance data is then used as input into the prediction mechanism, thus allowing derivation of a new predictive model.

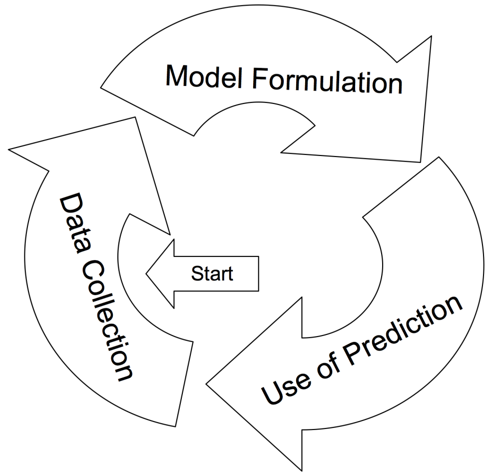

Figure 3.7: Prediction Cycle

This can either be a continuous process, dynamically reacting to environmental changes, or a static process, used once in order to determine an initial starting point. The DOE process and the subsequent development of an algorithm demonstrate that multi-linear models are sufficient for predicting a system configuration or informing a process that achieves target performance goals (specific results from this process are reported in chapter 4). The use of multi-linear regression models is most easily understood using the terminology of DOE and ANOVA techniques. This framework demonstrates which factors or parametric settings significantly contribute to predictive accuracy.

The power of cognitive radio is drawn from its ability to reconfigure in response to a change in the radio frequency environment or a change in application requirements. Central to developing any technique for intelligently reconfiguring the cognitive network is a solid understanding of how an individual radio's settings can affect its performance. DOE is set of tools and methods for determining cause and effect relationships within a system or process [2]. Traditionally, DOE has been used in the process industry to optimize product yield or to maximize production line efficiency. In our case, we use DOE to help determine which configuration of the CR's parameters will have the most positive impact on performance. DOE is ideally suited to answer questions of the form, "What is the best configuration of input factors or combination thereof to maximize an output or response?" Use of the DOE methodology requires a set of structured tests wherein permutations are made to the input parameters and the effects or responses of those changes are analyzed. Thus, DOE provides a method for understanding the relationships among input parameters and response metrics. The DOE process allows researchers to determine the significance of input factors acting alone, as well as in combination on the measured response. DOE makes no assumptions about how the various inputs interact or impact the outputs. This technique requires a set of experiments that produce adequate and statistically significant coverage of the experimental space. Mechanically, it relies on ANOVA to provide an assessment of the significance of the test results. The core statistical process at work is the calculation of the **F-test** [2]. This test compares the variance among the treatment means versus the variance of the individuals within the specific treatments. Another way of looking at the **F-test** is as a ratio of signal to noise.

The first step in using DOE is to identify the input variables and the responses. Each input variable has a number of levels. The input variable is varied along each of the levels and the result is measured. Tables 3.1 and 3.2 list input parameters, environmental variables and their settings. Each of the simulation runs evaluates the performance of the experimental system at each of the potential parametric settings. Table 3.3 is a list of the responses or metrics used to evaluate each mutation of the settings. One can independently look at the performance of any of the parameters (alone or in combination with other parameters) against any one of the metrics used to evaluate the system. A software suite developed by Stat-Ease was used to perform the DOE calculations [90]. This system also generates an equation for predicting a response given a set of input parameters. This equation can be used by a cognitive system to react to changes in environmental conditions or requirements. The following section illustrates to how to apply DOE and ANOVA and serves as a basis for later analysis.

Table 3.4: Factors and Responses for 802.11 Wireless Test Example

Factor	Units	Levels	Response
Frame Size	Bits	2048,18432	Latency
Bit Rate	Mbps	1,2	Latency
Transmit Power	mWatts	5,100	Latency

3.3.6.1 An Example

The first step in the DOE analysis is to identify those **factors** (i.e., inputs to an experiment) that will have an effect on the **response** (i.e., output of the experiment). Factors have different levels or values, for example, an 802.11 wireless card may have the capability to transmit at two different power settings. Specifically, in this example the factor, transmit power, has the levels of 32 and 100 mWatts. Whereas the response is a single value that represents a metric, observation, or measure. In this wireless card example, latency, the response, is measured across a noisy link at each of the levels of the factor. Table 3.4 shows a set of factors for an experiment wherein one determines which factors, or **interactions** between factors, have the most significant impact on latency. An interaction occurs when a factor at one level does not produce the same response at each level of another factor (i.e., latency is not consistent when power is fixed at 32mW and Bit Rate varies from 1 to 11Mbps). Once the factors, their levels, and the responses are determined you are ready to move on to the next step.

Next a set of experiments is run that iterate through all the combinations of factors at each of their levels. This wireless example requires eight experiments to encompass all of the possible configurations. Table 3.5 shows the set of experiments and the observed latency for each run. With a simple example such as this, you may be able to determine the best configuration through simple inspection of the results. However, when there are many factors, levels, and responses optimization by inspection becomes a time consuming, error prone and difficult task.

Table 3.5: Results from 802.11 Wireless Test Example

Run Order	Frame Size	Bit Rate	Transmit Power	Latency
1	2048	1	5	0.01430
2	18432	1	5	0.01257
3	2048	2	5	0.00660
4	18432	2	5	0.00570
5	2048	1	100	0.01453
6	18432	1	100	0.01276
7	2048	2	100	0.00660
8	18432	2	100	0.00580

The next step in the analysis is to make use of statistical methods to identify the those factors or interactions that impact the response, in this case latency. DOE makes use of ANOVA to determine which factors or interactions most impact the response. Rather than provide a primer on the statistics at work in the ANOVA, an interpretation of the results better serves the goal. The method used to calculate and build the ANOVA table is described in [2]. Table 3.6 is the ANOVA generated for the experimental runs given in Table 3.5. The sum of squares for the model and residual are shown in the first column of the ANOVA table. The next column is the degrees of freedom associated with the sum of squares. Next is the mean square, or the sum of squares divided by the degrees of freedom. The ratio of the mean squares of the model over the mean squares of the residual forms the next column, and is referred to as the **F-value**. The F-value is compared to the reference distribution for F, in order to determine the probability of observing this result due to error. In this example, we are using the reference distribution for a 95 percent confidence ratio. If you look closely at the table you will see that Frame Size and Bit Rate and their interaction are the factors that, according to their high F-value and their probability for error, are the most significant factors impacting latency (those factors and interactions with a **P Value Prob < F** of 0.1 or less are statistically significant).

Table 3.6: ANOVA for Latency

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	1.119E-4	6	1.865E-5	6472	<0.05
A-FrameSize	3.233E-6	1	3.233E-6	1122	<0.05
B-BitRate	1.082E-4	1	1.082E-4	37537	<0.05
C-TransmitPower	2.753E-8	1	2.75E-8	10	0.1992
AB	4.499E-07	1	4.499E-7	156	0.051
AC	9.643E-10	1	9.643E-10	0.335	0.666
BC	1.607E-8	1	1.607E-8	5.573	0.255
R^2: 0.99					

The model behind the ANOVA is a mathematical equation used to predict the response given a set of inputs. In the general case, the equation is of the form given in Equation 3.1. Where $\hat{Y}$ is the response and β_0 is the intercept and β_1 is the coefficient for the input factor, X_1. The larger the coefficient the greater the effect on the prediction. The equation for latency, after reducing the model to only the significant factors (FrameSize, DataRate, and their interaction) is given in Equation 3.2. To get an estimate of latency for a given configuration of the CR we need only plug values into this equation. For a frame size of 2048 and a bit rate of 2 Mbps (entered as 2,000,000 bits) one gets a value of 0.006585 seconds, which is very close to the observed response given in Table 3.5 (note, power is not included because the ANOVA did not indicate it as a significant factor). It follows that R^2, a measure of how well a regression line approximates the real data points, is 0.99 (see Table 3.6). Statistically speaking, the model for this simple example provides an almost perfect predictor.

$$\hat{Y} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_{12\ldots n} X_1 X_2 \ldots X_n \tag{3.1}$$

$$Lat = 0.0226 - 1.644\text{E-}7A - 7.948\text{E-}9B + 5.789\text{E-}14AB \tag{3.2}$$

Chapter 4 provides a summary of the most important results of the DOE and ANOVA process. The following section discusses the method used in development of the cognitive algorithm.

3.4 Algorithm Development

The algorithm for controlling reconfiguration of the system was designed to take advantage of the DOE analysis. The ANOVA tables and charts detailing the single and multi-factor interactions were the motivation for the design. The ANOVA tables indicate which **knobs** have the most significant impact on the response of interest. The thesis of this work is that a C/SDR can improve wireless performance by exploiting cross-layer parametric optimization in the presence of an active source of noise. Achieving this thesis then becomes the primary goal of the algorithm controlling the C/SDR. The following subsections describe the components of the algorithm as they relate to achieving this goal.

3.4.1 Algorithm Decomposition

The algorithm's description follows the framework outlined in Section 3.2.1 with a minor caveat. The thesis of this work did not require implementation of every component in the C/SDR framework. Those components that were integral to the pursuit of the thesis are described in detail below. Figure 3.8 gives a pictorial representation of the framework, as it relates to the implemented cognitive process.

- **Requirements Collection Component** - This component of the algorithm builds upon the **adverbs** and **adjectives** work completed during phase one of the research. The abstraction proposed earlier was simplified, as the utility of the adverbs abstraction was not the focus of this thesis. The adverbs abstraction is reduced to a **requirements vector**, wherein requirements to the system

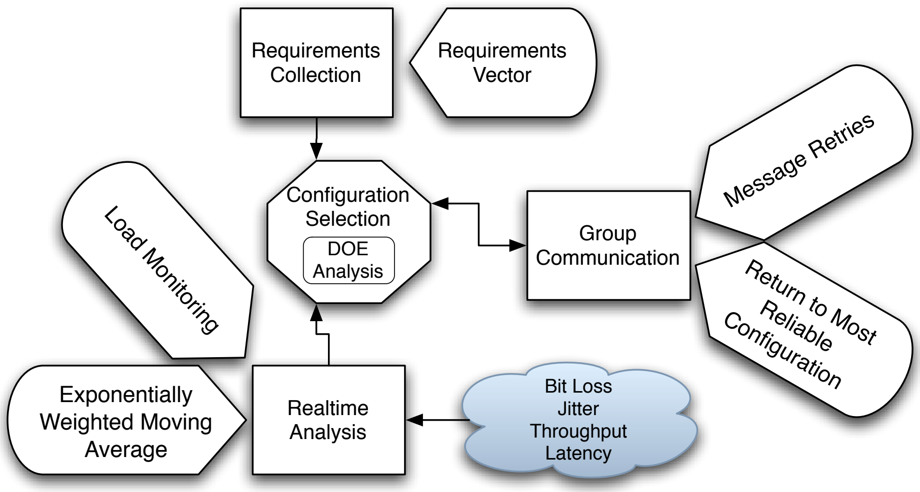

Figure 3.8: Components of the Cognitive Algorithm

are specified via a list of tolerances. This vector provides the needs for each application running on the system. The vector is composed of the worst case performance characteristics (e.g., tolerances) for each of the measured metrics and a threshold. The threshold is the trigger point for enacting the reconfiguration process. In the case where more than one application is providing its requirements, the individual application vectors are combined with the least restrictive tolerances discarded. For example, if one application could tolerate a latency of 10 seconds (e.g., file transfer), and another could only tolerate a latency of 10 milliseconds (e.g., VoIP), the least restrictive (in this case 10 seconds), is dropped and the more restrictive requirement retained.

- **Real-time Analysis Component** - This component's task is to monitor system performance. When a metric drops below its tolerance threshold a reconfiguration action may occur. Statistics are requested from the client node on a periodic basis. This period is called the **reconfiguration interval**. This component evaluates the performance of the system once at the beginning of each reconfiguration interval based on statistics gained during communication with the client node. This information is provided to the reconfiguration component to determine if a reconfiguration is warranted. Similar to those tools which track financial performance, an Exponential Moving Average (EMA) is used to indicate trends in system performance [87]. Rather than use a simple moving average, which gives equal weight to all data points, the EMA gives more weight to the most recent data points. Therefore, the EMA is able to respond more quickly to change than a simple moving average. An EMA takes a percentage of the current metric (i.e., throughput) and adds in the prior periods exponential moving average. For instance, suppose you wanted a 10% EMA. You would take the current value for throughput and multiply it by 10% then add that value to the prior periods EMA, EMA_Prev, multiplied by 90%. The formula for the calculation the EMA is given in Equation 3.3. The percentage used in this formula equates to time periods according to the formula provided in Equation 3.4.

$$EMA\_Current = (throughput\_current * 0.1) + (EMA\_Prev * (1 - 0.1)) \quad (3.3)$$

$$Percentage = 2/(Time\_Periods + 1) \quad (3.4)$$

- **Group Communication Component** - This component is charged with ensuring statistical and reconfiguration information is communicated between nodes. For reconfiguration, the configuration of the system is embedded in the 802.11 preamble. This standard preamble is always of the same format and sent at the 1 Mbps data rate. The use of the preamble for configuration alleviates the need for a message exchange dedicated to reconfiguration. However, in order to get performance information across the two communicating nodes, a statistics

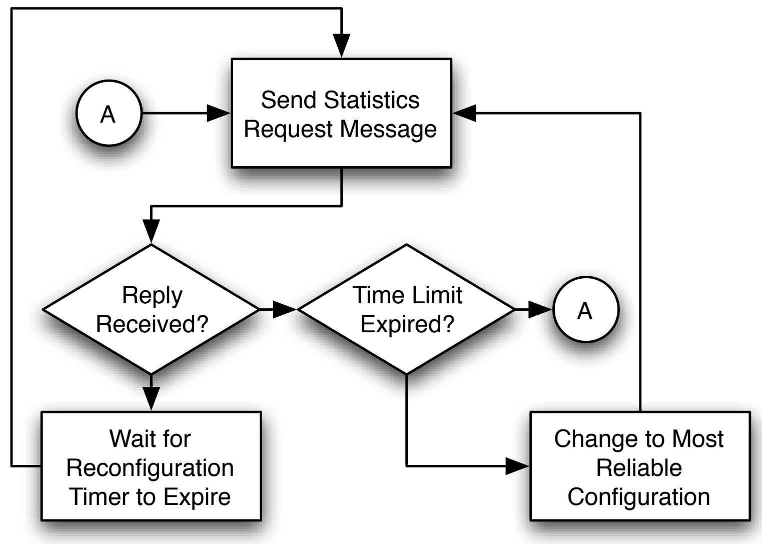

Figure 3.9: Message Exchange for Reconfiguration Cycle

message exchange is still required. This exchange is needed because the sending node has no knowledge of lost frames unless the receiver explicitly notifies it. This component uses two methods for ensuring that statistical information is exchanged in as reliable manner as feasible. Figure 3.9 illustrates this process. The first is a message retry mechanism. At the beginning of a reconfiguration interval, a statistics request message is sent from the server to the client. If this message is lost due to jamming the server will resend the message several times. If the response from the client is lost, the server will assume that the client response was lost and issue another request. If after several attempts, the reconfiguration continues to fail the server will fall back to its most **reliable** configuration and attempt the reconfiguration again. There are several well known formal methods for reliable message passing in a distributed system [24]. These methods would be considered if developing a production system.

- **Configuration Selection Component** - This component is responsible for determining the next configuration of the radio. This decision is based on performance information (from the realtime analysis component), and the requirements vector (from the requirements collection component). If the C/SDR is performing in accordance with the thresholds in the requirements vector then a configuration change is not required. However, there are secondary concerns. It makes sense when considering resource use and fairness to operate in a configuration that meets requirements, yet minimizes power output as well as time on the link (i.e., a most conservative configuration). By minimizing the time on the link and using the lowest possible power setting, a pair of nodes reduces the chance of interfering with other communicating nodes. DOE analysis provides the foundation for determining the next configuration (if the radio system is in a state where it is not meeting a performance goal). To determine the next configuration, the algorithm uses ANOVA tables to determine which factors most influence the metric of interest. For example, bit rate was the most influential single or multi-factor parameter impacting bit loss. This is evident by it having the highest **F-Value** (recall the DOE example in section 3.3.6.1). Thus, from the ANOVA table we are able to learn which factors most influence each metric. However, in order to know if the influence is positive or negative one must also look at **how** each factor or factor interaction influences the response. For example, ARQ has a significant impact on latency, however, this is a negative impact as ARQ increases latency on the link (due to the transmit of the acknowledgment frame for each data frame sent). This negative influence is apparent when looking at ARQ's effect on latency. A predictive model provides this information during the DOE process. The graphs for predicted latency given a change in bit rate and/or ARQ show whether an effect is positive or negative. The most significant graphs are discussed in detail in chapter 4. The algorithm uses a **greedy** approach to solving problems with a performance metric that is not meeting its specified goal. For example, if the system was reporting that latency is too high, the algorithm would start with the factor which most positively impacts latency and change it. This process is repeated, through secondary and tertiary factors, until the system either meets the performance goal or it is unable to change configuration to affect a positive change. Figure 3.10 illustrates this process.

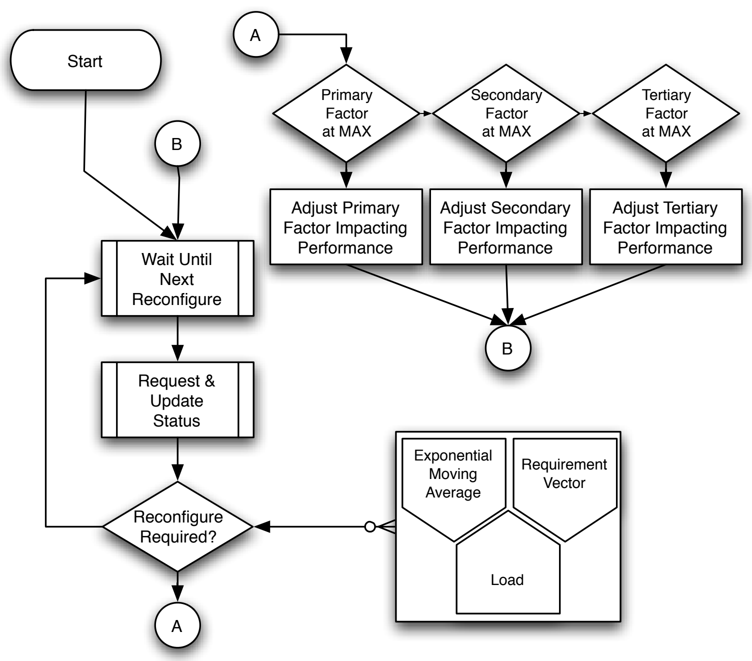

Figure 3.10: Configuration Selection Process

3.4.2 Metrics for Algorithm Evaluation

There are two facets to evaluating the reconfiguration algorithm. The first is general reporting on the characteristics of the reconfiguration algorithm (e.g., time to reconfigure). The set of metrics used to evaluate the reconfiguration algorithm is given in Table 3.7. The second is comparison of the reconfiguration algorithm of interest with other methods (e.g., most conservative configuration vs. neural net). In this research, evaluation is made against the best and worst performing static configurations (i.e.,

Table 3.7: Reconfiguration Metrics

Metric	Units
Reconfiguration Time	Seconds
Bit Loss	Percent
Latency	Seconds
Jitter	Seconds
Throughput	bps

reconfiguration is disabled). The specifics of the algorithm's evaluation are provided in chapter 4.

- **Reconfiguration Time** - This is a measure of how long it takes the C/SDR to change from one set of parametric settings to another. This metric is averaged across each run and reported for each reconfiguration interval.

Underlying the experimental analysis and design of the cognitive algorithm was the development of an experimental platform for the testing and evaluation of a C/SDR. The next section presents the development of the simulation platform used during this research.

3.5 Experimental Platform - Simulation in OPNET

The OPNET Modeler simulation environment was used to determine the effects of the input parameters on the responses and to evaluate the cognitive algorithm [89]. This software suite provides a rich and readily extensible network simulation and modeling environment. The following subsections present the setup and layout of the simulation environment, its limitations, and concludes with an overview of the simulation design.

3.5.1 Setup and Layout

The simulation consisted of two nodes communicating in the presence of a noise source. Figure 3.11 shows the layout of the two nodes in relationship to the noise source.

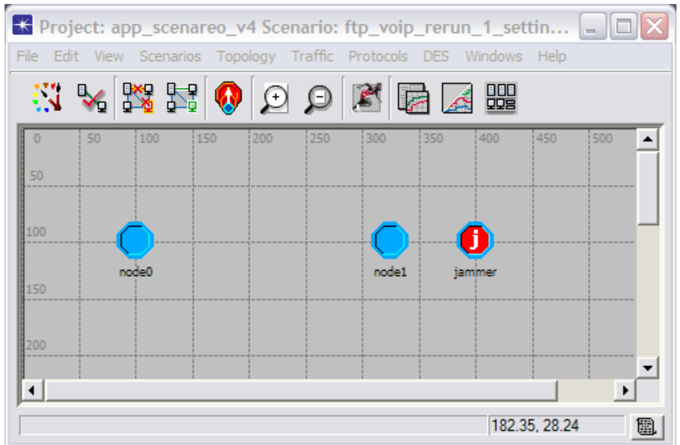

Figure 3.11: Simulation Layout

3.5.2 Limitations of the Simulation Platform

A general problem with many simulations is the lack of fidelity in the physical model. The complexities of wireless communication are such that it makes it very difficult to model the radio frequency environment with near perfect accuracy (e.g., multi-path, spurious noise sources). However, even without perfect fidelity in the physical model, OPNET is accurate enough to allow theoretical experimentation [89]. Ultimately, the algorithms and techniques developed and tested on the simulation platform would be implemented on a fielded radio test bed. The following text summarizes the specific limitations of the simulation.

- **Physical Model Limitations** - The physical model used by OPNET offers many of the features that one would expect of a commercial quality simulator. The radio pipeline accounts for background noise, attenuation due to distance of transmission, curvature effects of the Earth, and calculates effective signal to noise ratio (SNR) at the receiver. However, without additional software and geospacial modeling, one cannot calculate multi-path interference or attenuation due to obstructions. Also the stock radio pipeline allows a node to receive its transmissions. The OPNET pipeline allows successful reception of a frame if any power is received (no filter for receiver sensitivity). To fix these problems, the pipeline was modified to filter out energy levels that were below the sensitivity threshold of typical 802.11 equipment. Additionally, modifications were made so that transmissions from a node are not received by the same node.

- **Media Access Control Layer Limitations** - While OPNET provides a wireless networking module for the MAC layer, to obtain the flexibility that was required for interactions spanning protocol layers, it was necessary to develop a custom module. Additionally, during the development of the simulation platform, it was discovered that the protocol stack that shipped with OPNET was easily overwhelmed by the level of packet loss experienced when the jamming node was active. To surmount these two major limitations with the MAC and protocol stack, a custom model and set of analysis tools was developed.

3.5.3 Custom Simulation Model

This section briefly describes the development of the custom simulation model. The model is presented from the node level down to the media access control layer. OPNET uses a proprietary programming model that combines a graphical state machine, a vast library of predefined functions, and modified C code (proto-C). Slightly more than 7200 lines of custom code were generated during the development of the experimental platform.

3.5.3.1 Node Model

OPNET requires the creation of a node model, this node model depicts packet flow within the node. Figure 3.12 shows the node model created for the simulation platform. Data packets are generated by the file transfer and VoIP sources. Data packets flow to the MAC process (RMAC) where they are encapsulated for transmission and sent to the radio transmitter (xmt). Frames flow into the node via the receiver (rcv). Data frames

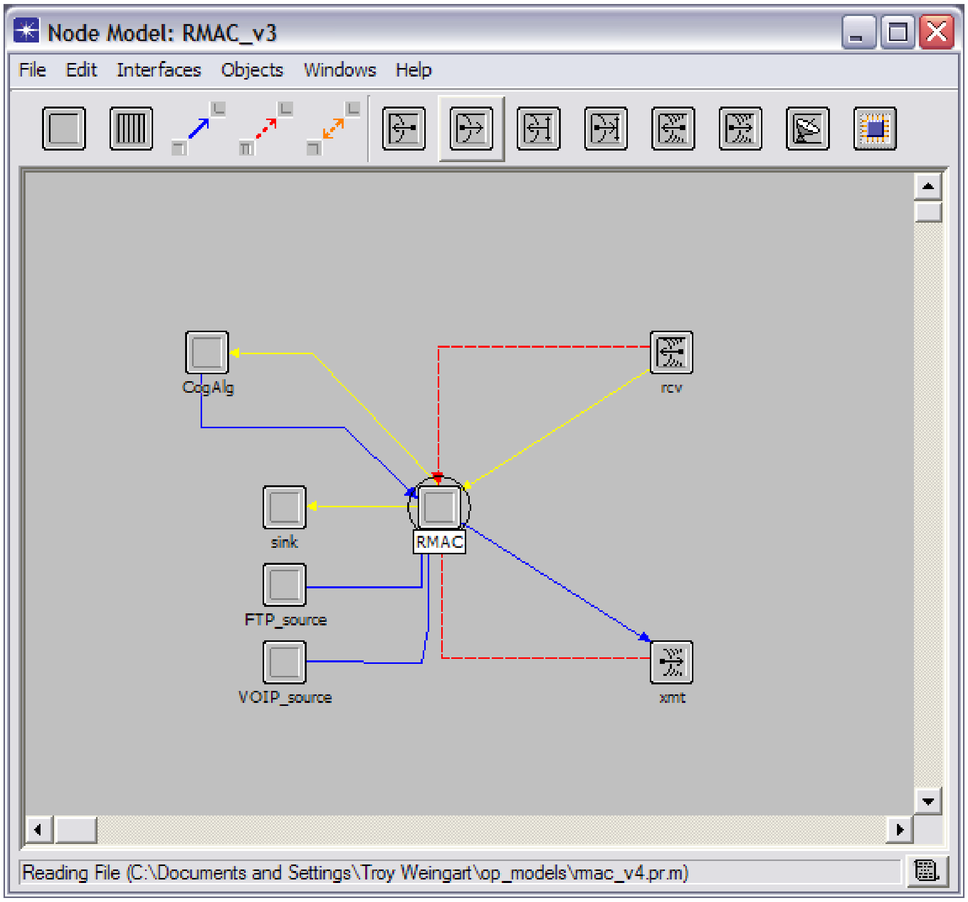

Figure 3.12: Node Model

are passed to a packet sink for statistic updates and management frames are sent to the cognitive process (CogAlg).

3.5.3.2 Cognitive and Statistics Process

The cognitive process is tasked with determining the radio's next configuration. Additionally, it manages performance metrics and responds to requests for statistic updates. The state machine is provided in Figure 3.13 for completeness. It is not important to understand the details of the process diagram, as the algorithm and its function are adequately described in Section 3.4. The cognitive process is divided between client and server functions. The server, or master node, is shown in the top half of the diagram.

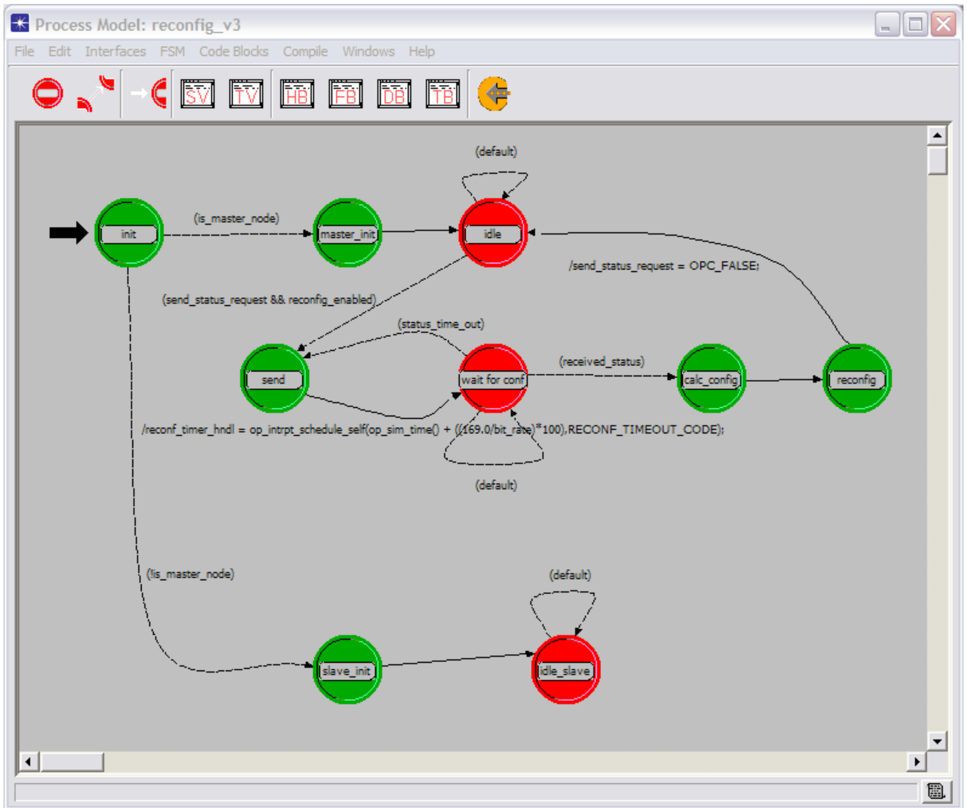

Figure 3.13: Cognitive Process

The client, or slave, is shown on the bottom. Green signifies a nonblocking state, while red signifies a conditional state.

3.5.3.3 Media Access Control Layer

MAC layer development was entirely driven by the limitations in the OPNET base modules. Carrier Sense Multiple Access (CSMA) was chosen as the core media access method in order to accurately mirror a future implementation in MultiMAC. The MAC allows one to change any of the radios settings on a per packet basis. The state diagram is included for completeness, but detailed description of the MACs operation is outside the scope of this document (see Figure 3.14).

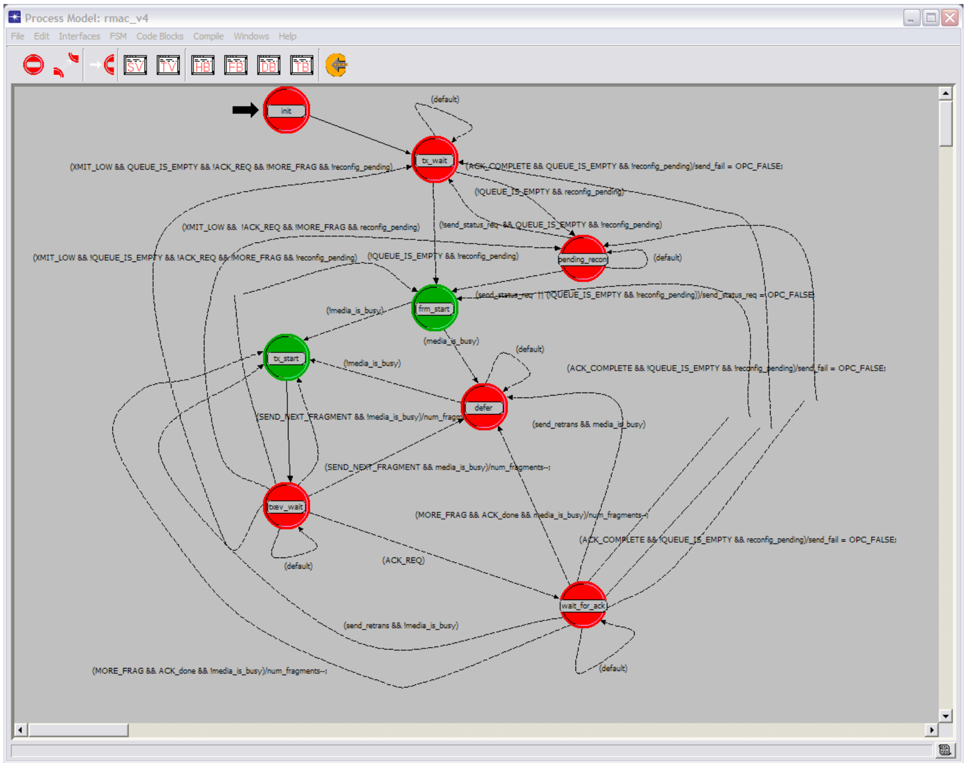

Figure 3.14: Media Access Control Layer

3.5.3.4 Tools

A set of post processing tools were created in Perl to distill and format the performance results for inclusion in the DOE software [90]. The following chapter covers the experimental results and their analysis.

Chapter 4

Simulation Results and Findings

The goals of this chapter are to guide the reader through the experimental parametric results and analyze the performance of the reconfiguration algorithm. The chapter is broken into two distinct parts. The first, presents an analysis of the key results obtained during the parametric experimentation phase of the research, highlighting those results which influenced the design of the cognitive algorithm. The second part presents an analysis of the algorithm used to tune the C/SDR. The intent in developing this algorithm is not to provide a provably optimal solution; but to posit an affective approach for dynamic reconfiguration of a C/SDR. This chapter begins with the experimental analysis.

4.1 Experimental Analysis

This section presents an analysis of the effects of changing radio parameters on performance. Emphasis is placed on those results that had an influence on the design of the reconfiguration algorithm. A set of experiments were run that encompassed all 2592 configurations of the C/SDR and performance data was gathered using the simulation platform. The following subsections report the results and findings. Presentation is organized as follows: (1) The average effect of parameter settings on performance. (2) The Analysis of Variance (ANOVA) tables. (3) The significant single and multi-factor effects. (4) And lastly, the predictive models.

Table 4.1: Internal Parameters

Factor	Levels	Layer
Selective Queueing	off/on	Network
Automatic Repeat Request (ARQ)	off/on	MAC
Framesize	2048,9216,18432 bits	MAC
Forward Error Correction	off/on	MAC
Bit Rate	1,2,5.5,11 Mbps	Physical
Transmit Power	5,32,100 mW	Physical

Table 4.2: External Parameters and Environmental Variables

Parameter	Settings
Jammer Noise Level	1,4,11 mW
Offered Load	0.5,1.5,5.5 Mbps

4.1.1 Average Effect of Parameter

The first step in the analysis is to identify the input variables (parameters) and the responses (metrics). Each input variable has a number of levels. The input variable is varied along each of the levels and the result is measured. Tables 4.1 and 4.2 list input parameters, environmental variables and their settings. Each of the simulation runs evaluates the performance of the experimental system at each of the potential parametric settings. Table 4.3 is a list of the metrics used to evaluate each mutation of the C/SDR's settings. One can independently look at the performance of any of the parameters (in combination with other parameters) against any one of the metrics used to evaluate the system.

Table 4.3: The Set of Metrics

Metric	Units
Bit Loss	Percent
Latency	Seconds
Jitter	Seconds
Throughput	bps

Recall that the initial approach for analysis of the data collected during the parametric experimentation phase was to use a simple average across all runs in order to determine a parameter settings effects (see Section 3.3.5). The average effect of a parameter is calculated across all possible configurations. For example, this technique allows for determining the effects of enabling or disabling forward error correction on throughput. This approach allows one to get a general sense of the effect of a parameter setting. Before a detailed discussion of the charts that follow it is important to understand the distinctions among the "perspectives" from which the performance metrics are collected.

Table 4.4: Perspectives on Performance Metrics

Perspective	Description
Higher Layer (HL)	Aggregate results viewed from above the MAC
File Transfer (FTP)	Individual results viewed by FTP from above the MAC
Voice over IP (VoIP)	Individual results viewed by VoIP from above the MAC
C/SCR MAC (RMAC)	Results on MAC layer with C/SDR extensions

The metrics of interest were analyzed from four distinct perspectives (see Table 4.4). When performance is reported from the perspective of the entire application layer, this is termed higher-layer (HL). HL is the aggregate performance of Voice over IP (VoIP) and file transfer (FTP) applications. Next performance is divided among the two applications, FTP and VoIP. FTP and VoIP traffic were selected due to their distinct tolerances for latency, jitter, throughput, and bit loss. The final perspective is that viewed from the MAC layer (RMAC). For the remainder of this document, unless explicitly stated, the reader should assume that reported results are from the higher layer perspective. The focus of the analysis will be on those results which improve the aggregate performance of the applications, however, for completeness the RMAC layer is analyzed as well. On the following pages are the charts detailing the average effect of a paramter setting on throughput. Presentation of the charts is followed by their analysis. The y-axis is throughput and the x-axis represents a change in parameter setting. Abbreviations in the charts are as follows, Selective Queueing (SelQ), Automatic Repeat Request (ARQ), Transmit (TX), Forward Error Correction (FEC).

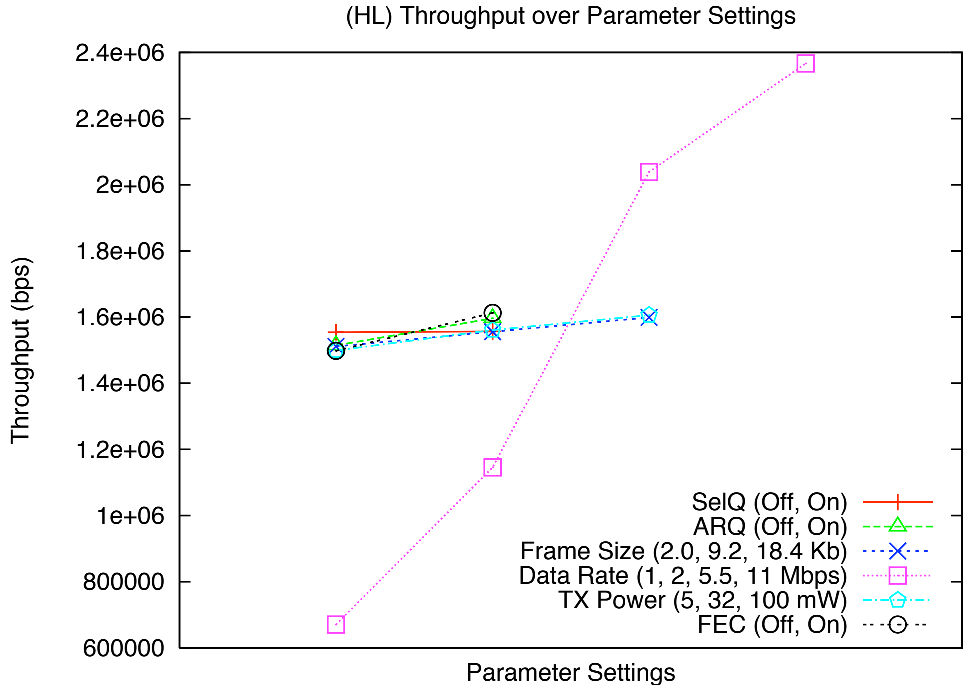

Figure 4.1: Average Effect of a Parameter on HL Throughput

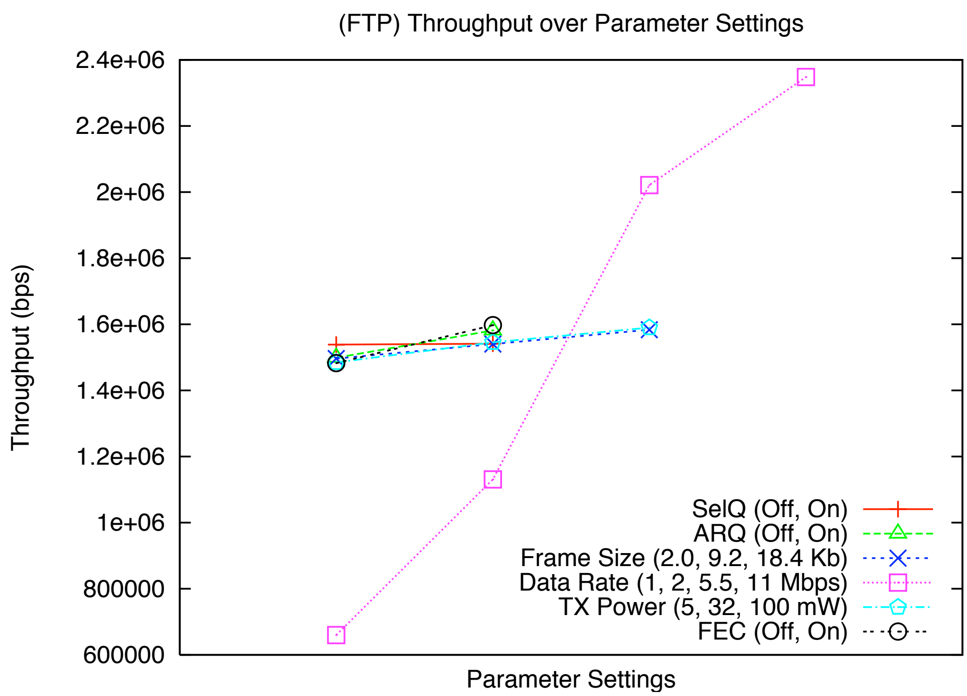

Figure 4.2: Average Effect of a Parameter on FTP Throughput

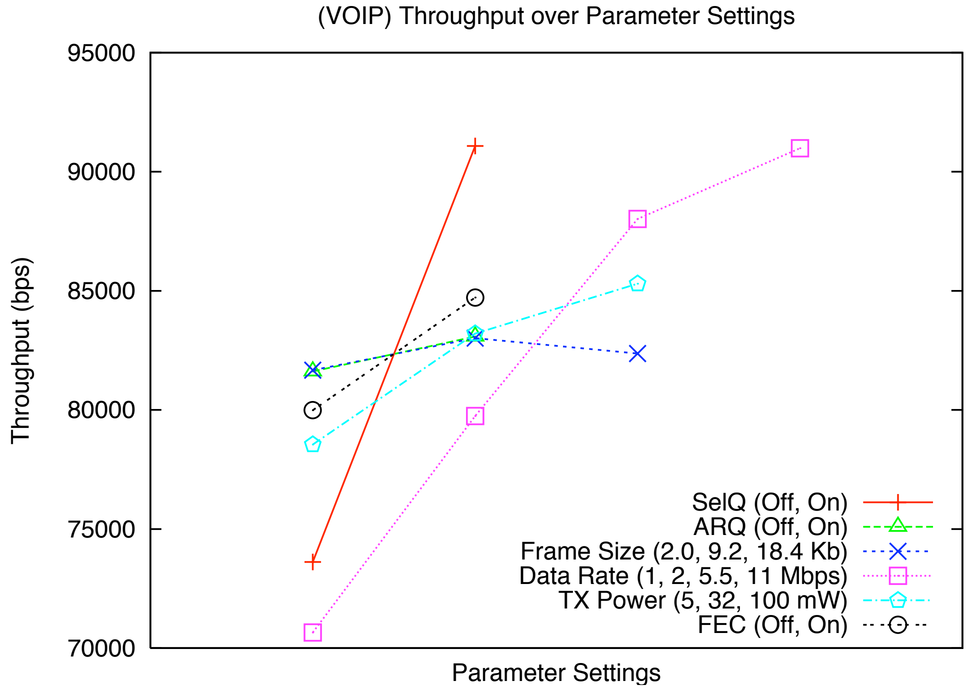

Figure 4.3: Average Effect of a Parameter on VoIP Throughput

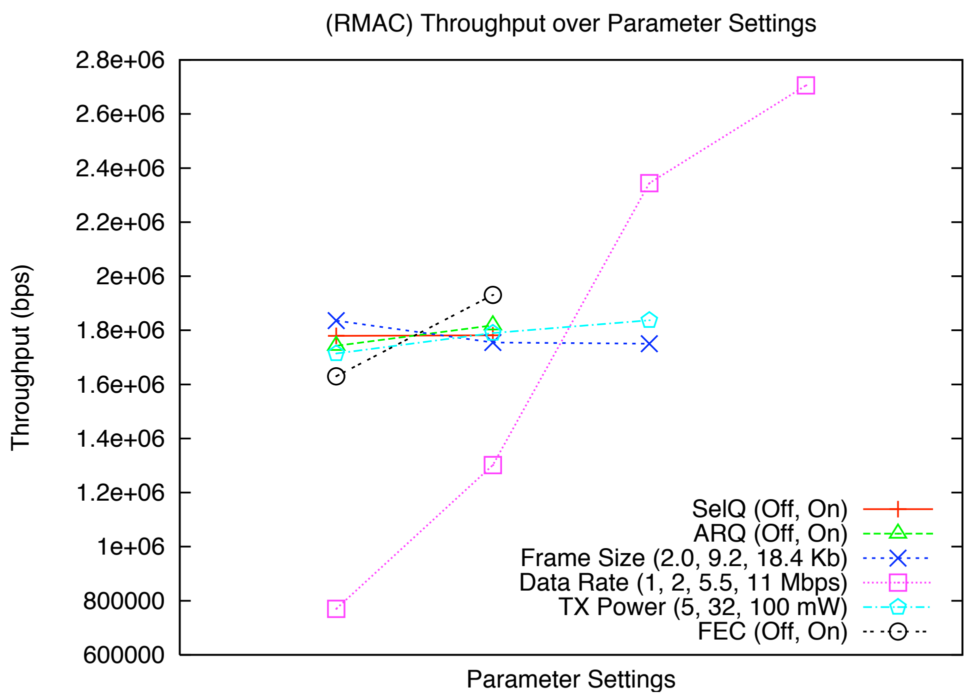

Figure 4.4: Average Effect of a Parameter on RMAC Throughput

The following is a summary of the major findings with respect to throughput (as shown in Figures 4.1, 4.2, 4.3, and 4.4).

(1) Data rate improves throughput in all cases and has the largest impact. It is interesting to note that the relative effect of the other parameters is minor. However, on average, the other parameters when enabled or increased, improve throughput (with a few exceptions that are detailed next).

(2) The factor that has the greatest impact on VOIP throughput is SelQ (when enabled SelQ enqueues VoIP frames at the front of the transmit queue). Additionally, SelQ has a negligible effect on FTP throughput. This is desirable as FTP performance is not adversely affected and VoIP benefits.

(3) Framesize has a negligible effect on VoIP throughput because the VoIP frame is not fragmented by the MAC (it is smaller than 2048 bits). The variance in VoIP throughput, when looking at frame size, is caused by interaction with the file transfer that is occurring at the same time.

(4) One would assume that a larger frame would improve RMAC throughput as it did at the application layer, however, because of the jammer, minimizing "air time" seems to offer better performance. This is because the frame is less likely to be hit by a jamming burst. Conversely, not fragmenting a frame at the MAC layer has a positive impact on application layer throughput.

On the following two pages are the charts for the average effect of a parameter setting on bit loss. Presentation of the charts is followed by their analysis. The y-axis is bit loss and the x-axis represents a change in parameter setting. Abbreviations in the charts are as follows, Selective Queueing (SelQ), Automatic Repeat Request (ARQ), Transmit (TX), and Forward Error Correction (FEC).

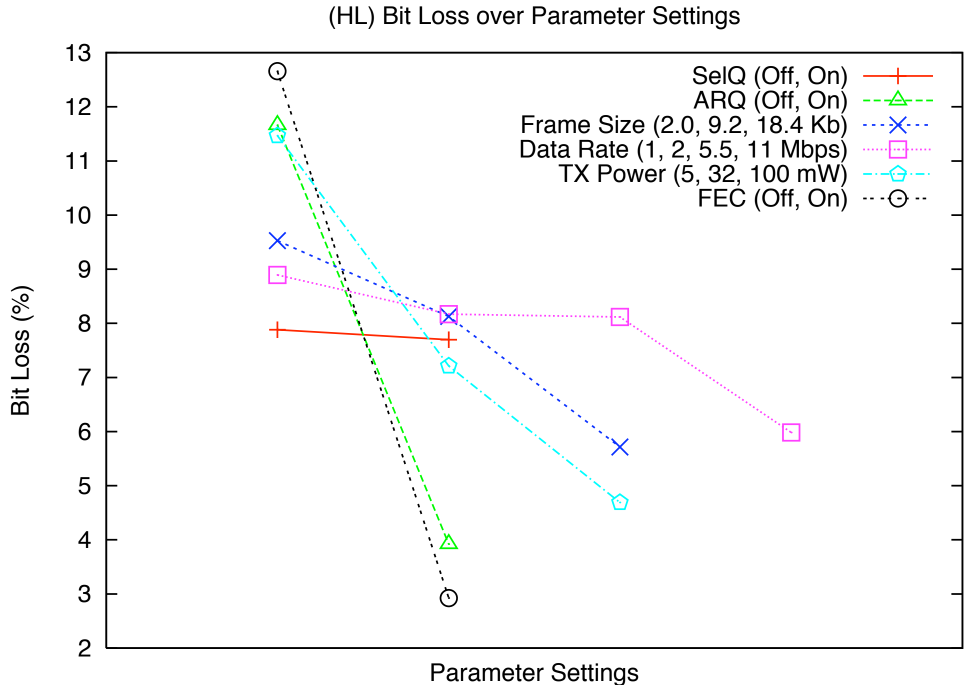

Figure 4.5: Average Effect of a Parameter on HL Bit Loss

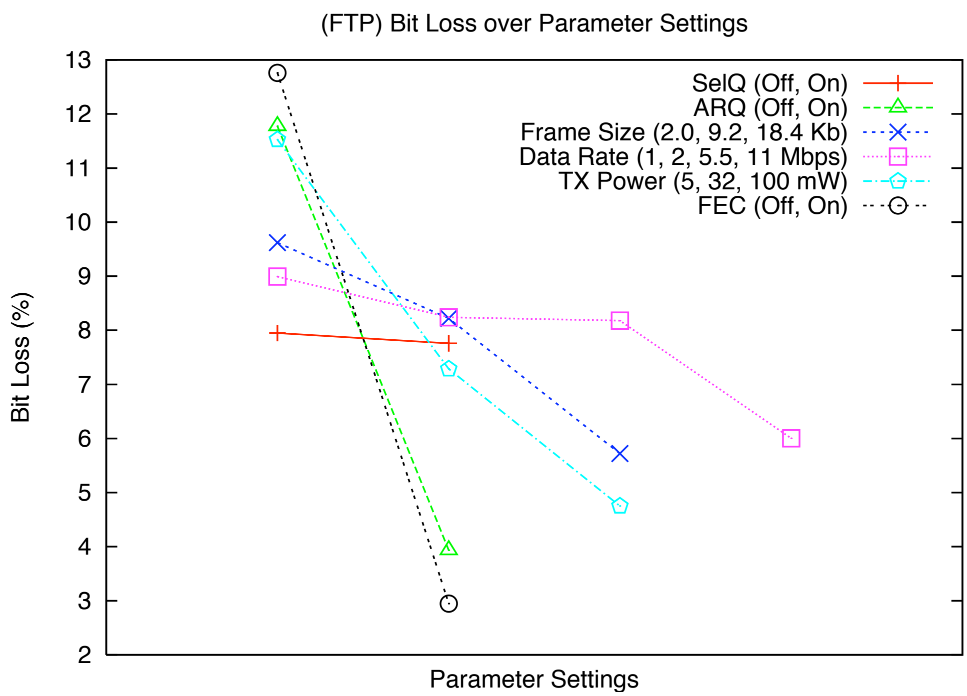

Figure 4.6: Average Effect of a Parameter on FTP Bit Loss

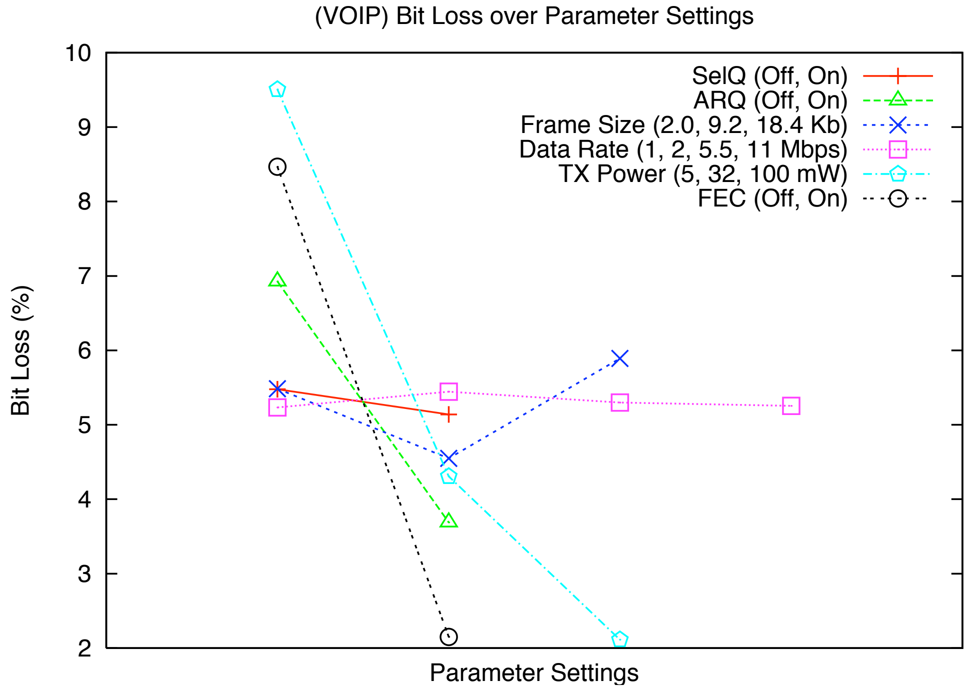

Figure 4.7: Average Effect of a Parameter on VoIP Bit Loss

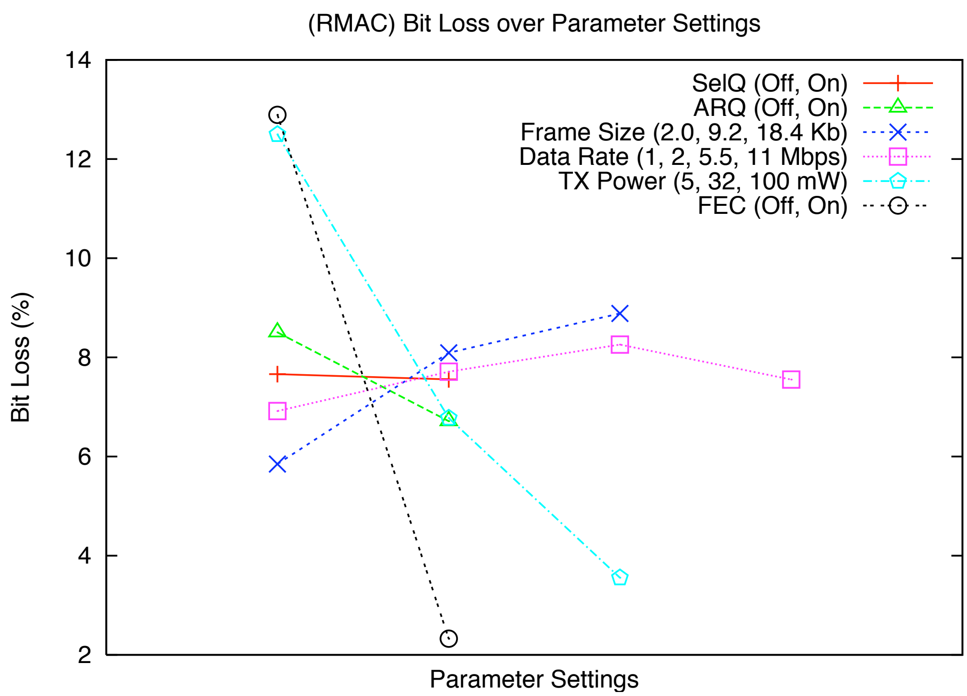

Figure 4.8: Average Effect of a Parameter on RMAC Bit Loss

The following is a summary of the major findings with respect to bit loss (as shown in Figures 4.5, 4.6, 4.7, and 4.8).

(1) FEC and TX Power are the two factors which have the greatest impact on bit loss. Additionally, ARQ has a positive impact, but to a lesser degree.

(2) An increase in FrameSize improves aggregate performance at the application layer (HL), however, there is an inverse relationship between the HL and MAC layer performance due to the interaction of "air time" and the overhead of fragmenting a frame. Therefore, it is advantageous to use a larger frame size and not fragment the frame at the MAC, rather than fragmenting in an attempt to decrease time on the noisy channel.

(3) Data rate improves bit loss at the application layer, but at the RMAC layer one sees a decline in performance due to the interaction of data rate and the associated change in modulation. The higher data rates use modulation schemes that are more susceptible to noise. This is not evident at the application layer due to interactions with the other parametric settings (the effect of modulation is balanced by other factors like ARQ, and XmitPower).

(4) SelQ has a very slight positive impact on VoIP bit loss and no appreciable effect otherwise.

On the following two pages are the charts for the average effect of a parameter setting on latency. Presentation of the charts is followed by their analysis. The y-axis is latency and the x-axis represents a change in parameter setting. Abbreviations in the charts are as follows, Selective Queueing (SelQ), Automatic Repeat Request (ARQ), Transmit (TX), and Forward Error Correction (FEC).

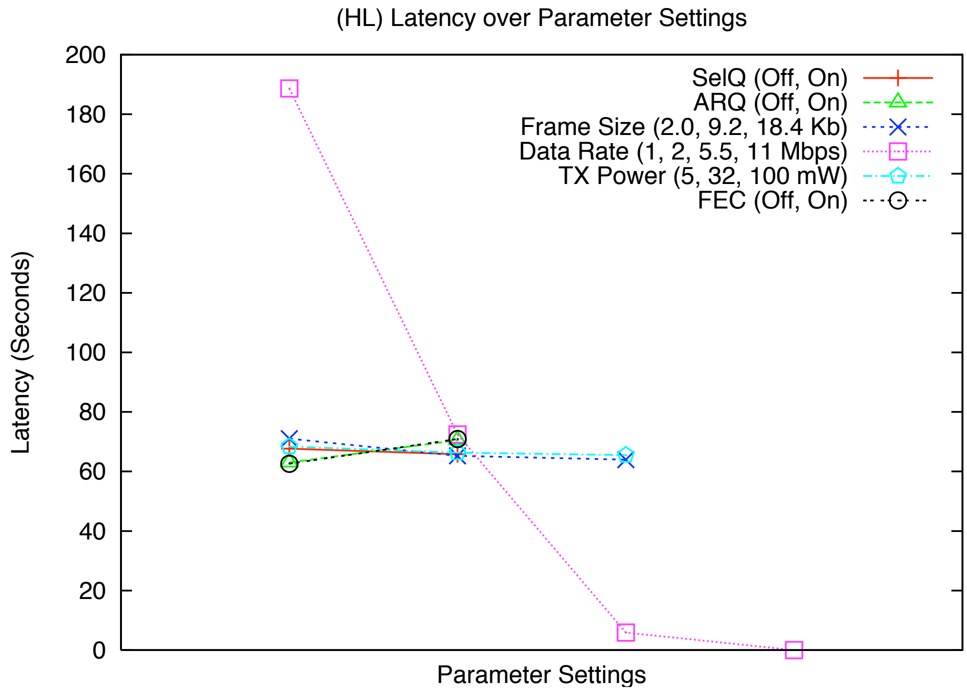

Figure 4.9: Average Effect of a Parameter on HL Latency

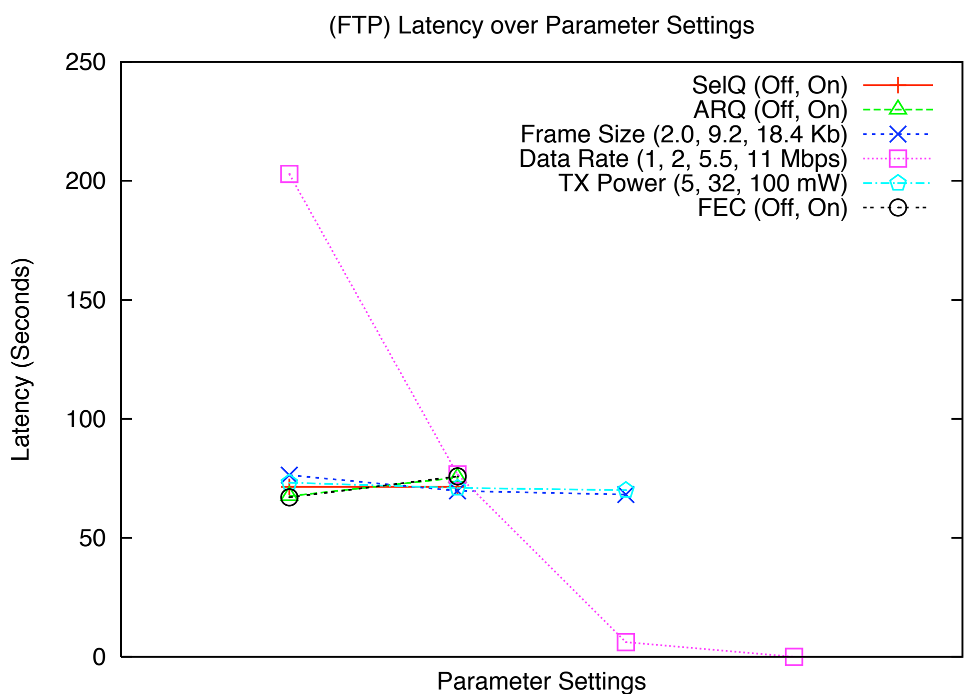

Figure 4.10: Average Effect of a Parameter on FTP Latency

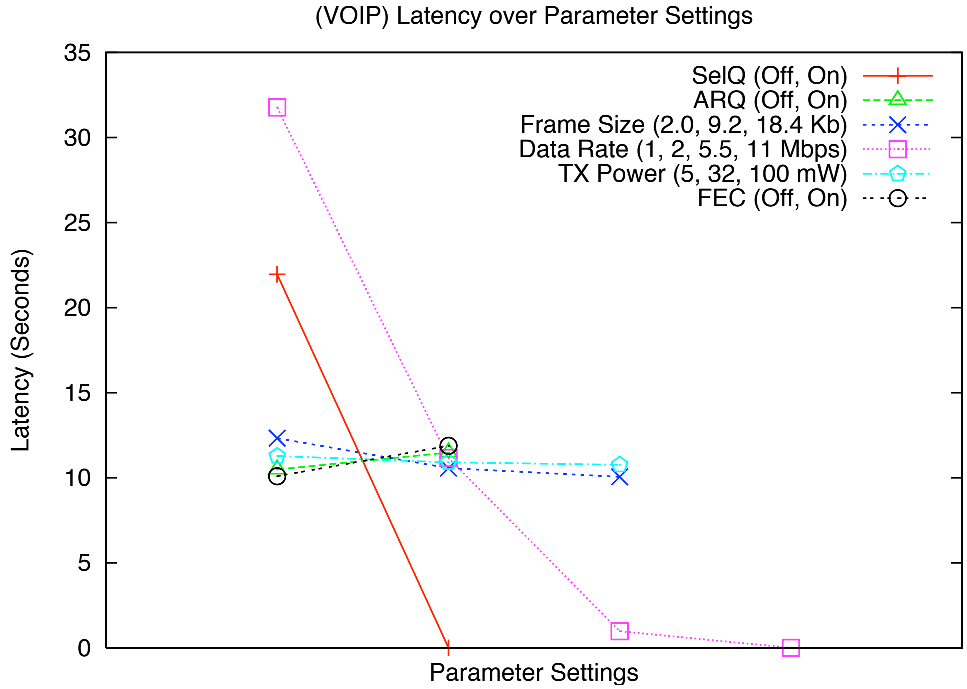

Figure 4.11: Average Effect of a Parameter on VoIP Latency

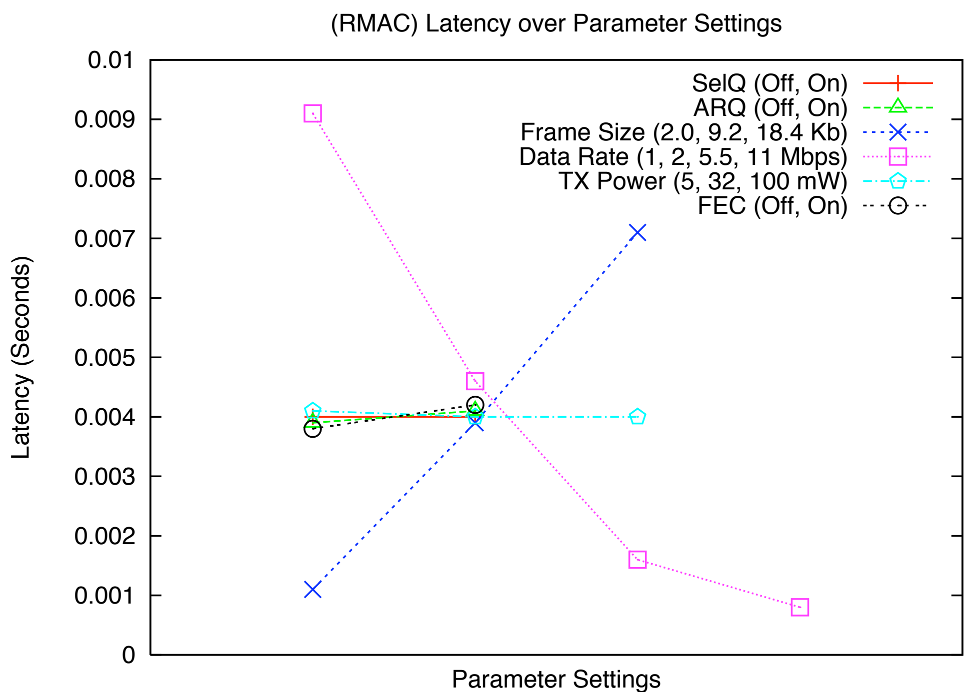

Figure 4.12: Average Effect of a Parameter on RMAC Latency

The following is a summary of the major findings with respect to latency (as shown in Figures 4.9, 4.10, 4.11, and 4.12).

(1) Data rate improves latency in all cases, and has the largest impact.

(2) SelQ, which was designed to improve VoIP latency, has the most significant positive effect on VoIP latency (dropping it two orders of magnitude). Additionally, it does not have an appreciable negative effect on FTP latency.

(3) FrameSize, as one would expect, increases latency at the MAC layer. Conversely, due to the overhead of fragmentation, the larger FrameSize slightly improves latency at the application layer.

(4) ARQ has a slight negative impact on latency due to the overhead in the acknowledgment exchange.

(5) TX Power has no appreciable effect on latency.

(6) FEC increases latency due to the overhead in the parity bits attached to each frame.

On the following two pages are the charts for the average effect of a parameter setting on jitter. Presentation of the charts is followed by their analysis. The y-axis is jitter and the x-axis represents a change in parameter setting. Abbreviations in the charts are as follows, Selective Queueing (SelQ), Automatic Repeat Request (ARQ), Transmit (TX), and Forward Error Correction (FEC).

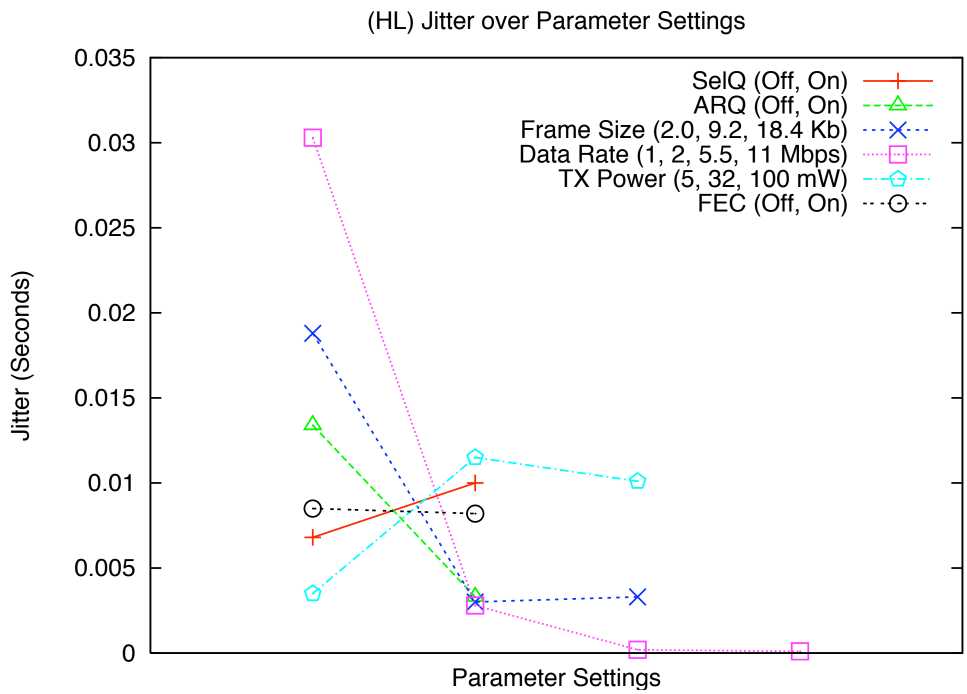

Figure 4.13: Average Effect of a Parameter on HL Jitter

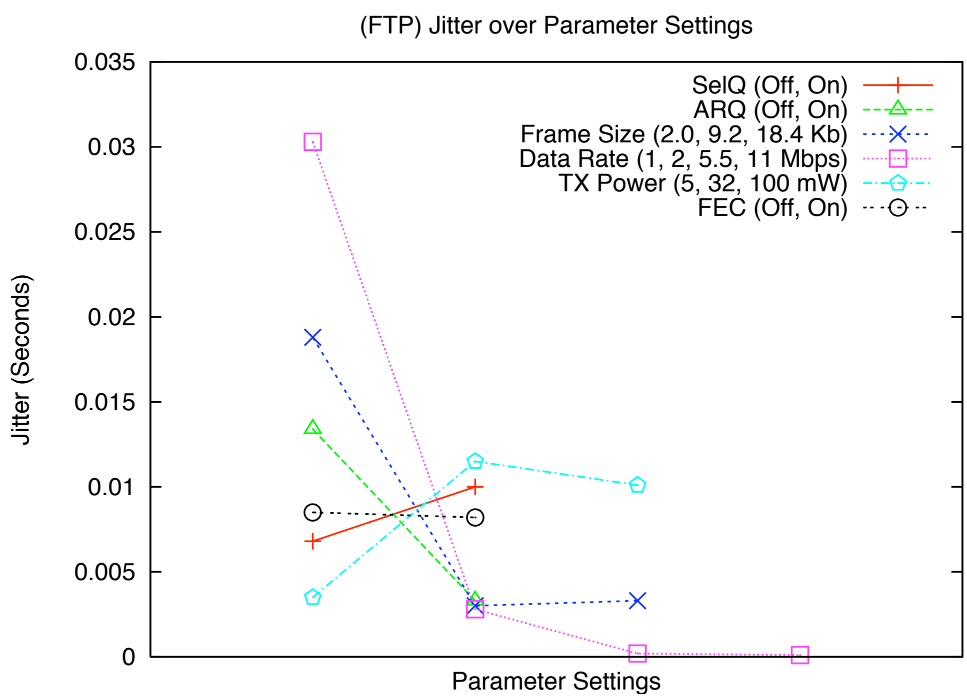

Figure 4.14: Average Effect of a Parameter on FTP Jitter

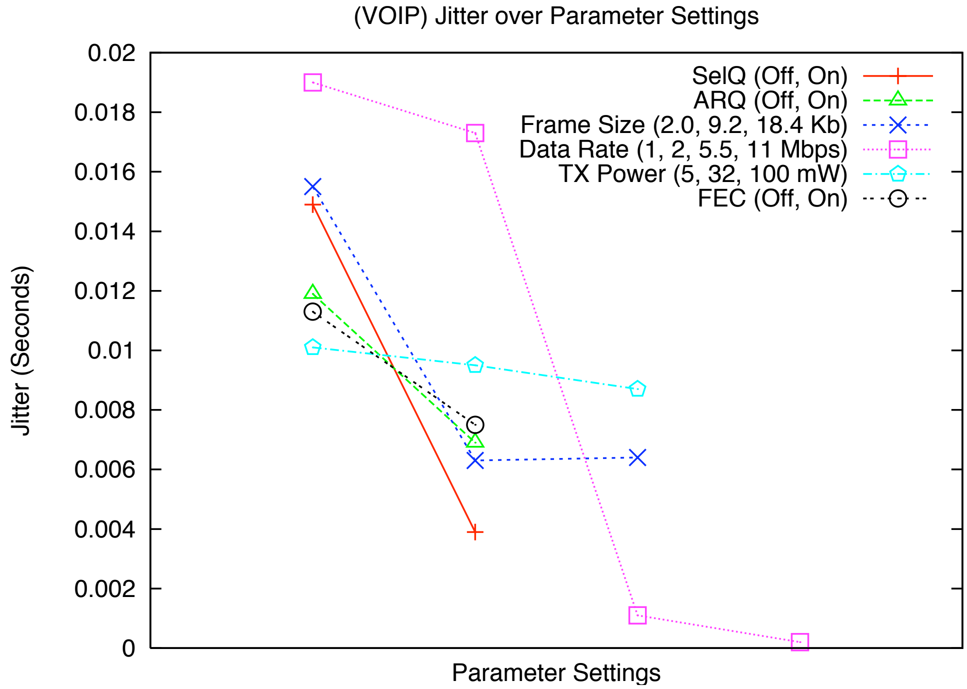

Figure 4.15: Average Effect of a Parameter on VoIP Jitter

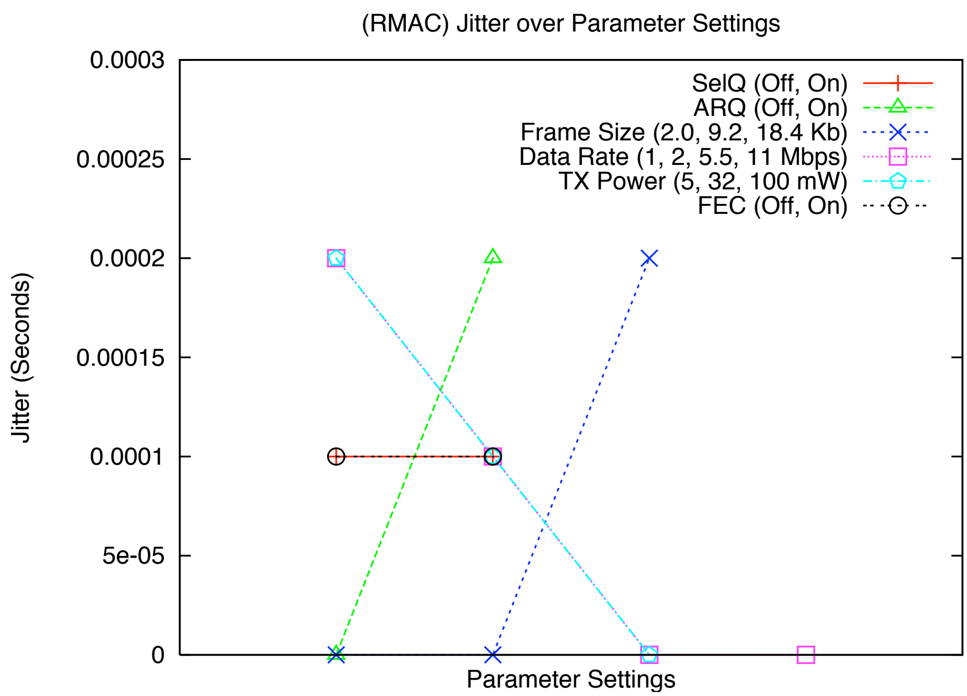

Figure 4.16: Average Effect of a Parameter on RMAC Jitter

The following is a summary of the major findings with respect to jitter (as shown in Figures 4.13, 4.14, 4.15, and 4.16).

(1) Data rate improves jitter in all cases and has the largest impact. The other factors have inconsistent interactions and impact on jitter.

(2) SelQ and FEC have negligible effect on jitter with respect to the MAC layer. However, at the application layer their effect is mixed with SelQ increasing jitter and FEC decreasing it. The performance of FEC can be attributed to its better bit loss performance. FEC when enabled causes fewer frames to be retransmitted. SelQ, as designed, improves VoIP jitter.

(3) FrameSize and ARQ have a negative effect at the MAC layer due to variance in frame length due to unpredictability in fragmentation and a variable number of retries and/or acknowledgments. FrameSize has a mixed effect at the application layer. Aggregate and FTP thoughput improve when FrameSize is increased. VoIP data, because it is not fragmented at the MAC, experiences an interaction with FTP frames, causing the larger frame size to improve VoIP jitter.

(4) TX Power improves jitter at the RMAC and VoIP layers, but increases it with respect to aggregate and FTP performance.

4.1.1.1 Average Effect of a Parameter - Findings

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm.

- **Throughput** - Increasing date rate is the single and most important factor in improving throughput. The effects of the other parameter settings are negligible, with SelQ being the exception. SelQ should be enabled permanently as it does not adversely effect aggregate throughput and is very beneficial for VoIP.

However, if streaming video were also running on the server one would want to selectively enable and disable SelQ as it could potentially adversely impact other latency sensitive applications. The cognitive algorithm will focus on data rate as remedy for a drop in throughput.

- **Bit Loss** - Enabling FEC and increasing TX Power have a large positive impact on bit loss with data rate increases having a slight positive impact. Additionally, ARQ enabling improves bit loss, but to a lesser degree. Also, SelQ when enabled, has no appreciable negative effect on bit loss, reaffirming that it should always be enabled. It would make sense to target FEC, TX Power, and data rate as the remedies for bit loss problems.

- **Latency** - Increasing data rate is the single and most effective factor impacting latency in all cases. However, when combined with SelQ, VoIP latency is dramatically improved (strengthening the case for leaving SelQ permanently enabled). It appears that the remedies for latency mirror those for throughput.

- **Jitter** - Jitter was the most problematic of the metrics to analyze because it is a measure of variance in latency. However, there was one constant, an increase in data rate positively affects jitter. Again, like throughput and latency, the reconfiguration algorithm will focus on data rate as a remedy for jitter problems.

The common thread throughout this analysis is that a data rate increase has a positive impact on every performance metric. Bit loss is the one metric that has a different remedy strategy, enabling of FEC/ARQ and increasing transmit power. Additionally SelQ, has a very beneficial effect on VoIP performance and does not have an adverse effect on file transfer. For the application suite used in this thesis, SelQ should be enabled. Large FrameSize (no fragmentation) offered the best performance at the application layer, making a strong case for avoiding fragmentation of packets.

These findings, once confirmed by Design of Experiments (DOE), will be key factors influencing the development of the reconfiguration algorithm. However, in moving forward one must realize that average effect of a parameter analysis lacks specificity and ignores potentially harmful parametric interactions. Therefore, it is important to validate these findings with ANOVA and DOE techniques.

4.1.2 Analysis of Variance

The experimental data used in determining the average effect of a parameter was also used in the DOE analysis. This data was imported into Stat-Ease, a DOE software support suite [90]. The following presents an ANOVA with respect to throughput, bit loss, latency, and jitter, from each of the four perspectives (HL, FTP, VoIP, RMAC). During this phase particular attention is given to those C/SDR parameters that have statistically significant influence on the response of interest. Recall from the previous chapter that high F-values indicate relative impact on the response (see Section 3.3.6.1 for a detailed description of this process). ANOVA allows one to make a quantitative assessment as to which parameters have the greatest impact on the response of interest. As indicated in chapter 3, the reconfiguration algorithm uses the three most significant factors impacting a response in order to react to changes in performance, therefore deconstruction of the ANOVA tables is limited to the three most significant factors impacting the response. Additionally, those findings are highlighted that correlate or contradict the results from the average effect of a setting analysis.

On the following four pages are the ANOVA tables for the average effect of a parameter setting on throughput. Presentation of the tables is followed by their analysis. For completeness, those parameters that are external to the C/SDR are included in the ANOVA (load, jammer power). Abbreviations in the tables are as follows J_Power (Jammer Power), SelectQ (Selective Queueing), ARQ (Automatic Repeat Request), and xmitPower (Transmit Power).

Table 4.5: ANOVA for HL Throughput

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	1.79E+16	56	3.20E+14	8113.99	< 0.0001
A-Load	8.71E+15	1	8.71E+15	221156.3	< 0.0001
AF	5.29E+15	3	1.76E+15	44744.87	< 0.0001
F-DataRate	3.58E+15	3	1.19E+15	30315.49	< 0.0001
A^2	1.63E+14	1	1.63E+14	4146.65	< 0.0001
H-FEC	2.58E+13	1	2.58E+13	654.4	< 0.0001
G-xmitPower	1.32E+13	1	1.32E+13	335.94	< 0.0001
D-ARQ	1.32E+13	1	1.32E+13	334.42	< 0.0001
E-FrameSize	9.85E+12	1	9.85E+12	250.02	< 0.0001
FH	2.53E+13	3	8.44E+12	214.43	< 0.0001
AE	8.32E+12	1	8.32E+12	211.33	< 0.0001
DH	7.17E+12	1	7.17E+12	182.07	< 0.0001
B-J_Power	7.04E+12	1	7.04E+12	178.85	< 0.0001
AH	6.83E+12	1	6.83E+12	173.37	< 0.0001
GH	6.07E+12	1	6.07E+12	154.05	< 0.0001
AD	6.05E+12	1	6.05E+12	153.54	< 0.0001
AG	4.97E+12	1	4.97E+12	126.23	< 0.0001
AB	2.68E+12	1	2.68E+12	68.1	< 0.0001
G^2	1.63E+12	1	1.63E+12	41.28	< 0.0001
BH	1.61E+12	1	1.61E+12	40.79	< 0.0001
DF	4.32E+12	3	1.44E+12	36.57	< 0.0001
DE	1.42E+12	1	1.42E+12	36.05	< 0.0001
EF	3.49E+12	3	1.16E+12	29.52	< 0.0001
BG	8.07E+11	1	8.07E+11	20.48	< 0.0001
B^2	7.35E+11	1	7.35E+11	18.66	< 0.0001
BF	1.59E+12	3	5.30E+11	13.46	< 0.0001
EH	3.62E+11	1	3.62E+11	9.19	0.0024
FG	8.12E+11	3	2.71E+11	6.87	0.0001
BD	2.07E+11	1	2.07E+11	5.26	0.0219
E^2	6.06E+10	1	6.06E+10	1.54	0.2149
C-SelectQ	1.26E+10	1	1.26E+10	0.32	0.5715
CH	7.34E+09	1	7.34E+09	0.19	0.6660
EG	6.79E+09	1	6.79E+09	0.17	0.6780
AC	6.42E+09	1	6.42E+09	0.16	0.6864
DG	5.79E+09	1	5.79E+09	0.15	0.7015
CD	4.27E+09	1	4.27E+09	0.11	0.7421
CF	5.74E+09	3	1.91E+09	0.05	0.9858
CE	1.55E+09	1	1.55E+09	0.04	0.8425
BC	8.49E+08	1	8.49E+08	0.02	0.8833
CG	5.76E+08	1	5.76E+08	0.01	0.9038
BE	4.76E+08	1	4.76E+08	0.01	0.9124
R^2: 0.983					

Table 4.6: ANOVA for FTP Throughput

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	1.79E+16	56	3.19E+14	8150.93	< 0.0001
A-Load	8.77E+15	1	8.77E+15	223715.33	< 0.0001
AF	5.25E+15	3	1.75E+15	44676.1	< 0.0001
F-DataRate	3.55E+15	3	1.18E+15	30215.18	< 0.0001
A^2	1.64E+14	1	1.64E+14	4180.66	< 0.0001
H-FEC	2.55E+13	1	2.55E+13	651.35	< 0.0001
D-ARQ	1.32E+13	1	1.32E+13	336.58	< 0.0001
G-xmitPower	1.29E+13	1	1.29E+13	330.4	< 0.0001
E-FrameSize	9.87E+12	1	9.87E+12	251.89	< 0.0001
FH	2.49E+13	3	8.29E+12	211.54	< 0.0001
AE	8.25E+12	1	8.25E+12	210.57	< 0.0001
DH	7.22E+12	1	7.22E+12	184.26	< 0.0001
B-J_Power	6.89E+12	1	6.89E+12	175.77	< 0.0001
AH	6.85E+12	1	6.85E+12	174.94	< 0.0001
AD	6.03E+12	1	6.03E+12	153.78	< 0.0001
GH	5.94E+12	1	5.94E+12	151.71	< 0.0001
AG	5.01E+12	1	5.01E+12	127.85	< 0.0001
AB	2.70E+12	1	2.70E+12	68.8	< 0.0001
G^2	1.57E+12	1	1.57E+12	40.18	< 0.0001
BH	1.56E+12	1	1.56E+12	39.87	< 0.0001
DF	4.28E+12	3	1.43E+12	36.44	< 0.0001
DE	1.37E+12	1	1.37E+12	34.97	< 0.0001
EF	3.49E+12	3	1.16E+12	29.7	< 0.0001
BG	7.99E+11	1	7.99E+11	20.4	< 0.0001
B^2	7.23E+11	1	7.23E+11	18.45	< 0.0001
BF	1.59E+12	3	5.29E+11	13.49	< 0.0001
EH	3.43E+11	1	3.43E+11	8.76	0.0031
FG	7.96E+11	3	2.65E+11	6.77	0.0001
BD	2.01E+11	1	2.01E+11	5.14	0.0235
E^2	5.93E+10	1	5.93E+10	1.51	0.2186
C-SelectQ	1.32E+10	1	1.32E+10	0.34	0.5622
CH	7.78E+09	1	7.78E+09	0.2	0.6559
EG	6.43E+09	1	6.43E+09	0.16	0.6855
AC	6.34E+09	1	6.34E+09	0.16	0.6876
DG	5.68E+09	1	5.68E+09	0.15	0.7033
CD	3.31E+09	1	3.31E+09	0.08	0.7713
CF	5.61E+09	3	1.87E+09	0.05	0.9862
CE	1.42E+09	1	1.42E+09	0.04	0.8488
BC	1.14E+09	1	1.14E+09	0.03	0.8645
CG	7.36E+08	1	7.36E+08	0.02	0.8910
BE	6.31E+08	1	6.31E+08	0.02	0.8990
R^2: 0.983					

Table 4.7: ANOVA for VoIP Throughput

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	3.42E+12	56	6.10E+10	630.27	< 0.0001
A-Load	5.98E+11	1	5.98E+11	6177.01	< 0.0001
C-SelectQ	5.93E+11	1	5.93E+11	6123.27	< 0.0001
AC	5.74E+11	1	5.74E+11	5924.69	< 0.0001
CF	5.68E+11	3	1.89E+11	1956.44	< 0.0001
F-DataRate	4.87E+11	3	1.62E+11	1675.51	< 0.0001
AF	3.82E+11	3	1.27E+11	1313.18	< 0.0001
G-xmitPower	4.95E+10	1	4.95E+10	511.31	< 0.0001
H-FEC	4.35E+10	1	4.35E+10	448.81	< 0.0001
B-J_Power	1.95E+10	1	1.95E+10	201.14	< 0.0001
GH	1.83E+10	1	1.83E+10	188.98	< 0.0001
G^2	1.25E+10	1	1.25E+10	128.58	< 0.0001
FH	3.45E+10	3	1.15E+10	118.63	< 0.0001
A^2	7.61E+09	1	7.61E+09	78.6	< 0.0001
BH	5.50E+09	1	5.50E+09	56.76	< 0.0001
D-ARQ	4.16E+09	1	4.16E+09	42.95	< 0.0001
CH	2.74E+09	1	2.74E+09	28.27	< 0.0001
B^2	1.88E+09	1	1.88E+09	19.41	< 0.0001
E^2	1.85E+09	1	1.85E+09	19.06	< 0.0001
CE	1.83E+09	1	1.83E+09	18.93	< 0.0001
DE	1.73E+09	1	1.73E+09	17.84	< 0.0001
FG	3.85E+09	3	1.28E+09	13.25	< 0.0001
AH	1.27E+09	1	1.27E+09	13.12	0.0003
CG	1.07E+09	1	1.07E+09	11.09	0.0009
BC	5.94E+08	1	5.94E+08	6.13	0.0133
AG	5.75E+08	1	5.75E+08	5.94	0.0148
E-FrameSize	4.88E+08	1	4.88E+08	5.04	0.0248
EF	1.40E+09	3	4.66E+08	4.81	0.0024
BD	4.47E+08	1	4.47E+08	4.62	0.0317
DF	1.04E+09	3	3.47E+08	3.58	0.0133
DG	1.96E+08	1	1.96E+08	2.02	0.1549
EH	1.46E+08	1	1.46E+08	1.51	0.2194
AB	1.19E+08	1	1.19E+08	1.23	0.2678
DH	7.05E+07	1	7.05E+07	0.73	0.3936
AD	6.29E+07	1	6.29E+07	0.65	0.4203
EG	2.24E+07	1	2.24E+07	0.23	0.6303
CD	2.14E+07	1	2.14E+07	0.22	0.6386
BG	1.98E+07	1	1.98E+07	0.2	0.6508
BF	2.04E+07	3	6.81E+06	0.07	0.9758
AE	4.19E+06	1	4.19E+06	0.04	0.8353
BE	4.03E+06	1	4.03E+06	0.04	0.8384
R^2: 0.821					

Table 4.8: ANOVA for RMAC Throughput

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	2.35E+16	56	4.20E+14	8227.10	< 0.0001
A-Load	1.13E+16	1	1.13E+16	220485.53	< 0.0001
AF	6.93E+15	3	2.31E+15	45210.70	< 0.0001
F-DataRate	4.71E+15	3	1.57E+15	30757.05	< 0.0001
A^2	2.12E+14	1	2.12E+14	4151.85	< 0.0001
H-FEC	1.76E+14	1	1.76E+14	3450.72	< 0.0001
AH	5.86E+13	1	5.86E+13	1146.84	< 0.0001
FH	7.70E+13	3	2.57E+13	502.54	< 0.0001
G-xmitPower	1.70E+13	1	1.70E+13	333.57	< 0.0001
D-ARQ	1.13E+13	1	1.13E+13	221.27	< 0.0001
E-FrameSize	8.67E+12	1	8.67E+12	169.78	< 0.0001
GH	7.43E+12	1	7.43E+12	145.52	< 0.0001
AG	6.25E+12	1	6.25E+12	122.34	< 0.0001
B-J_Power	5.83E+12	1	5.83E+12	114.12	< 0.0001
DE	5.03E+12	1	5.03E+12	98.45	< 0.0001
DH	4.45E+12	1	4.45E+12	87.21	< 0.0001
EF	1.33E+13	3	4.43E+12	86.82	< 0.0001
BH	3.67E+12	1	3.67E+12	71.88	< 0.0001
AB	3.58E+12	1	3.58E+12	70.09	< 0.0001
AD	3.32E+12	1	3.32E+12	65.03	< 0.0001
E^2	3.23E+12	1	3.23E+12	63.18	< 0.0001
G^2	2.78E+12	1	2.78E+12	54.37	< 0.0001
AE	1.99E+12	1	1.99E+12	38.96	< 0.0001
B^2	1.03E+12	1	1.03E+12	20.21	< 0.0001
BD	9.58E+11	1	9.58E+11	18.76	< 0.0001
EH	7.81E+11	1	7.81E+11	15.29	< 0.0001
BF	1.65E+12	3	5.50E+11	10.78	< 0.0001
DF	1.53E+12	3	5.11E+11	10.01	< 0.0001
FG	1.31E+12	3	4.37E+11	8.55	< 0.0001
BG	3.67E+11	1	3.67E+11	7.18	0.0074
BE	2.39E+11	1	2.39E+11	4.68	0.0306
EG	4.62E+10	1	4.62E+10	0.91	0.3414
DG	1.87E+10	1	1.87E+10	0.37	0.5456
C-SelectQ	3.48E+09	1	3.48E+09	0.07	0.7942
CH	2.98E+09	1	2.98E+09	0.06	0.8092
AC	1.16E+09	1	1.16E+09	0.02	0.8800
BC	1.02E+09	1	1.02E+09	0.02	0.8875
CG	5.99E+08	1	5.99E+08	0.01	0.9138
CD	3.78E+08	1	3.78E+08	0.01	0.9314
CF	3.41E+08	3	1.14E+08	0.00	0.9999
CE	2.68E+05	1	2.68E+05	0.00	0.9982
R^2: 0.984					

The following is a summary of the major findings with respect to throughput (as shown in Tables 4.5, 4.6, 4.7, and 4.8).

(1) All C/SDR parameters have statistical significance with respect to their effect on throughput, with SelQ being the exception (its p-value probably indicates that is not a significant factor affecting throughput in any measure but VoIP). This correlates well with earlier assertions.

(2) Data rate is quantitatively confirmed to have the greatest impact on throughput in all cases but VoIP, where SelQ is most influential.

(3) FEC is the second most influential factor followed closely by TX Power and ARQ.

On the following four pages are the ANOVA tables for the average effect of a parameter setting on bit loss. Presentation of the tables is followed by their analysis. For completeness, those parameters that are external to the C/SDR are included in the ANOVA (load, jammer power). Abbreviations in the tables are as follows J_Power (Jammer Power), SelectQ (Selective Queueing), ARQ (Automatic Repeat Request), and xmitPower (Transmit Power).

Table 4.9: ANOVA for HL Bit Loss

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	5.47E+01	56	9.77E-01	1.97E+02	< 0.0001
H-FEC	1.84E+01	1	1.84E+01	3.71E+03	< 0.0001
D-ARQ	1.16E+01	1	1.16E+01	2.35E+03	< 0.0001
G-xmitPower	5.18E+00	1	5.18E+00	1.04E+03	< 0.0001
DH	2.74E+00	1	2.74E+00	5.52E+02	< 0.0001
E-FrameSize	1.92E+00	1	1.92E+00	3.86E+02	< 0.0001
B-J_Power	1.78E+00	1	1.78E+00	3.59E+02	< 0.0001
GH	1.64E+00	1	1.64E+00	3.31E+02	< 0.0001
G^2	9.01E-01	1	9.01E-01	1.82E+02	< 0.0001
DF	2.67E+00	3	8.90E-01	1.79E+02	< 0.0001
FH	2.48E+00	3	8.28E-01	1.67E+02	< 0.0001
EF	1.71E+00	3	5.72E-01	1.15E+02	< 0.0001
DE	3.81E-01	1	3.81E-01	7.68E+01	< 0.0001
BH	3.64E-01	1	3.64E-01	7.34E+01	< 0.0001
F-DataRate	9.20E-01	3	3.07E-01	6.18E+01	< 0.0001
B^2	2.53E-01	1	2.53E-01	5.10E+01	< 0.0001
FG	5.99E-01	3	2.00E-01	4.03E+01	< 0.0001
AH	1.91E-01	1	1.91E-01	3.84E+01	< 0.0001
AD	1.88E-01	1	1.88E-01	3.79E+01	< 0.0001
DG	9.68E-02	1	9.68E-02	1.95E+01	< 0.0001
AB	9.47E-02	1	9.47E-02	1.91E+01	< 0.0001
A-Load	6.64E-02	1	6.64E-02	1.34E+01	0.0003
BF	1.98E-01	3	6.59E-02	1.33E+01	< 0.0001
EG	5.23E-02	1	5.23E-02	1.05E+01	0.0012
AE	3.97E-02	1	3.97E-02	7.99E+00	0.0047
BD	3.13E-02	1	3.13E-02	6.30E+00	0.0121
AG	2.72E-02	1	2.72E-02	5.48E+00	0.0192
AF	7.86E-02	3	2.62E-02	5.28E+00	0.0012
A^2	2.34E-02	1	2.34E-02	4.71E+00	0.0300
E^2	1.24E-02	1	1.24E-02	2.49E+00	0.1144
C-SelectQ	6.77E-03	1	6.77E-03	1.36E+00	0.2429
CH	4.98E-03	1	4.98E-03	1.00E+00	0.3165
EH	4.98E-03	1	4.98E-03	1.00E+00	0.3166
CE	1.09E-03	1	1.09E-03	2.21E-01	0.6387
CF	3.23E-03	3	1.08E-03	2.17E-01	0.8849
CD	1.06E-03	1	1.06E-03	2.13E-01	0.6444
BC	9.81E-04	1	9.81E-04	1.98E-01	0.6566
AC	7.04E-04	1	7.04E-04	1.42E-01	0.7064
BE	1.91E-04	1	1.91E-04	3.84E-02	0.8446
CG	1.42E-04	1	1.42E-04	2.86E-02	0.8658
BG	5.40E-06	1	5.40E-06	1.09E-03	0.9737
R^2: 0.59					

Table 4.10: ANOVA for FTP Bit Loss

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	5.58E+01	56	9.96E-01	1.94E+02	< 0.0001
H-FEC	1.87E+01	1	1.87E+01	3.65E+03	< 0.0001
D-ARQ	1.20E+01	1	1.20E+01	2.33E+03	< 0.0001
G-xmitPower	5.18E+00	1	5.18E+00	1.01E+03	< 0.0001
DH	2.83E+00	1	2.83E+00	5.50E+02	< 0.0001
E-FrameSize	2.01E+00	1	2.01E+00	3.91E+02	< 0.0001
B-J_Power	1.78E+00	1	1.78E+00	3.47E+02	< 0.0001
GH	1.64E+00	1	1.64E+00	3.19E+02	< 0.0001
DF	2.75E+00	3	9.16E-01	1.78E+02	< 0.0001
G^2	8.90E-01	1	8.90E-01	1.73E+02	< 0.0001
FH	2.49E+00	3	8.31E-01	1.62E+02	< 0.0001
EF	1.77E+00	3	5.92E-01	1.15E+02	< 0.0001
DE	4.18E-01	1	4.18E-01	8.13E+01	< 0.0001
BH	3.64E-01	1	3.64E-01	7.09E+01	< 0.0001
F-DataRate	9.68E-01	3	3.23E-01	6.28E+01	< 0.0001
B^2	2.53E-01	1	2.53E-01	4.92E+01	< 0.0001
AH	2.16E-01	1	2.16E-01	4.20E+01	< 0.0001
FG	6.06E-01	3	2.02E-01	3.93E+01	< 0.0001
AD	1.59E-01	1	1.59E-01	3.10E+01	< 0.0001
DG	9.63E-02	1	9.63E-02	1.88E+01	< 0.0001
AB	9.60E-02	1	9.60E-02	1.87E+01	< 0.0001
A-Load	9.16E-02	1	9.16E-02	1.78E+01	< 0.0001
BF	2.02E-01	3	6.73E-02	1.31E+01	< 0.0001
EG	5.24E-02	1	5.24E-02	1.02E+01	0.0014
AE	4.91E-02	1	4.91E-02	9.57E+00	0.0020
A^2	3.61E-02	1	3.61E-02	7.03E+00	0.0080
BD	3.02E-02	1	3.02E-02	5.89E+00	0.0153
AG	2.72E-02	1	2.72E-02	5.30E+00	0.0214
AF	6.93E-02	3	2.31E-02	4.50E+00	0.0037
E^2	1.59E-02	1	1.59E-02	3.09E+00	0.0788
C-SelectQ	7.22E-03	1	7.22E-03	1.41E+00	0.2358
EH	5.55E-03	1	5.55E-03	1.08E+00	0.2986
CH	5.31E-03	1	5.31E-03	1.03E+00	0.3093
BC	1.30E-03	1	1.30E-03	2.53E-01	0.6151
CF	3.21E-03	3	1.07E-03	2.08E-01	0.8908
CE	9.62E-04	1	9.62E-04	1.87E-01	0.6652
CD	6.34E-04	1	6.34E-04	1.23E-01	0.7254
AC	5.24E-04	1	5.24E-04	1.02E-01	0.7495
CG	2.37E-04	1	2.37E-04	4.61E-02	0.8300
BE	2.02E-04	1	2.02E-04	3.93E-02	0.8428
BG	8.97E-09	1	8.97E-09	1.75E-06	0.9989
R^2: 0.58					

Table 4.11: ANOVA for VoIP Bit Loss

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	2.98E+01	56	5.33E-01	1.84E+02	< 0.0001
H-FEC	7.76E+00	1	7.76E+00	2.68E+03	< 0.0001
G-xmitPower	5.89E+00	1	5.89E+00	2.03E+03	< 0.0001
GH	2.06E+00	1	2.06E+00	7.10E+02	< 0.0001
D-ARQ	2.04E+00	1	2.04E+00	7.03E+02	< 0.0001
B-J_Power	1.98E+00	1	1.98E+00	6.82E+02	< 0.0001
G^2	1.59E+00	1	1.59E+00	5.49E+02	< 0.0001
FH	2.51E+00	3	8.38E-01	2.89E+02	< 0.0001
CD	6.44E-01	1	6.44E-01	2.22E+02	< 0.0001
AD	5.59E-01	1	5.59E-01	1.93E+02	< 0.0001
BH	5.39E-01	1	5.39E-01	1.86E+02	< 0.0001
DE	3.18E-01	1	3.18E-01	1.10E+02	< 0.0001
DF	8.28E-01	3	2.76E-01	9.53E+01	< 0.0001
B^2	2.50E-01	1	2.50E-01	8.64E+01	< 0.0001
A-Load	2.28E-01	1	2.28E-01	7.86E+01	< 0.0001
E^2	2.14E-01	1	2.14E-01	7.38E+01	< 0.0001
AE	2.13E-01	1	2.13E-01	7.35E+01	< 0.0001
FG	4.76E-01	3	1.59E-01	5.48E+01	< 0.0001
DH	1.24E-01	1	1.24E-01	4.29E+01	< 0.0001
CF	3.57E-01	3	1.19E-01	4.11E+01	< 0.0001
AF	3.20E-01	3	1.07E-01	3.69E+01	< 0.0001
EH	1.02E-01	1	1.02E-01	3.53E+01	< 0.0001
AC	9.85E-02	1	9.85E-02	3.40E+01	< 0.0001
BC	9.19E-02	1	9.19E-02	3.17E+01	< 0.0001
DG	7.98E-02	1	7.98E-02	2.76E+01	< 0.0001
EF	2.16E-01	3	7.21E-02	2.49E+01	< 0.0001
BD	5.74E-02	1	5.74E-02	1.98E+01	< 0.0001
CG	5.55E-02	1	5.55E-02	1.92E+01	< 0.0001
AH	5.48E-02	1	5.48E-02	1.89E+01	< 0.0001
E-FrameSize	3.23E-02	1	3.23E-02	1.12E+01	0.0008
EG	2.28E-02	1	2.28E-02	7.89E+00	0.0050
AB	2.22E-02	1	2.22E-02	7.65E+00	0.0057
C-SelectQ	2.21E-02	1	2.21E-02	7.63E+00	0.0058
BG	1.56E-02	1	1.56E-02	5.37E+00	0.0205
BF	3.37E-02	3	1.12E-02	3.88E+00	0.0087
A^2	9.74E-03	1	9.74E-03	3.36E+00	0.0668
AG	7.36E-03	1	7.36E-03	2.54E+00	0.1110
CH	4.19E-03	1	4.19E-03	1.45E+00	0.2292
F-DataRate	5.38E-03	3	1.79E-03	6.19E-01	0.6024
CE	1.32E-03	1	1.32E-03	4.56E-01	0.4997
BE	4.75E-04	1	4.75E-04	1.64E-01	0.6854
R^2: 0.57					

Table 4.12: ANOVA for RMAC Bit Loss

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	5.64E+01	56	1.01E+00	2.45E+02	< 0.0001
H-FEC	2.17E+01	1	2.17E+01	5.29E+03	< 0.0001
G-xmitPower	8.96E+00	1	8.96E+00	2.18E+03	< 0.0001
B-J_Power	3.57E+00	1	3.57E+00	8.68E+02	< 0.0001
GH	3.09E+00	1	3.09E+00	7.53E+02	< 0.0001
EH	1.76E+00	1	1.76E+00	4.28E+02	< 0.0001
G^2	1.68E+00	1	1.68E+00	4.08E+02	< 0.0001
FH	4.03E+00	3	1.34E+00	3.27E+02	< 0.0001
E-FrameSize	1.14E+00	1	1.14E+00	2.79E+02	< 0.0001
DE	1.13E+00	1	1.13E+00	2.75E+02	< 0.0001
BH	9.91E-01	1	9.91E-01	2.41E+02	< 0.0001
AD	9.37E-01	1	9.37E-01	2.28E+02	< 0.0001
D-ARQ	6.24E-01	1	6.24E-01	1.52E+02	< 0.0001
DF	1.85E+00	3	6.16E-01	1.50E+02	< 0.0001
AH	5.28E-01	1	5.28E-01	1.29E+02	< 0.0001
B^2	4.12E-01	1	4.12E-01	1.00E+02	< 0.0001
A-Load	3.90E-01	1	3.90E-01	9.49E+01	< 0.0001
EG	3.04E-01	1	3.04E-01	7.41E+01	< 0.0001
EF	8.26E-01	3	2.75E-01	6.70E+01	< 0.0001
BE	2.74E-01	1	2.74E-01	6.67E+01	< 0.0001
FG	7.92E-01	3	2.64E-01	6.43E+01	< 0.0001
AB	2.35E-01	1	2.35E-01	5.73E+01	< 0.0001
AG	1.75E-01	1	1.75E-01	4.25E+01	< 0.0001
DG	1.44E-01	1	1.44E-01	3.50E+01	< 0.0001
E^2	1.43E-01	1	1.43E-01	3.49E+01	< 0.0001
BD	1.04E-01	1	1.04E-01	2.53E+01	< 0.0001
AF	2.18E-01	3	7.28E-02	1.77E+01	< 0.0001
F-DataRate	1.78E-01	3	5.93E-02	1.44E+01	< 0.0001
A^2	5.46E-02	1	5.46E-02	1.33E+01	0.0003
BF	1.57E-01	3	5.22E-02	1.27E+01	< 0.0001
AE	7.91E-03	1	7.91E-03	1.93E+00	0.1652
DH	2.45E-03	1	2.45E-03	5.96E-01	0.4401
CH	2.11E-03	1	2.11E-03	5.14E-01	0.4735
C-SelectQ	2.09E-03	1	2.09E-03	5.09E-01	0.4755
BC	1.21E-03	1	1.21E-03	2.95E-01	0.5873
BG	7.73E-04	1	7.73E-04	1.88E-01	0.6645
CE	3.95E-04	1	3.95E-04	9.61E-02	0.7565
CF	3.45E-04	3	1.15E-04	2.80E-02	0.9937
CG	9.18E-05	1	9.18E-05	2.24E-02	0.8812
CD	1.24E-05	1	1.24E-05	3.02E-03	0.9562
AC	1.04E-06	1	1.04E-06	2.53E-04	0.9873
R^2: 0.64					

The following is a summary of the major findings with respect to bit loss (as shown in Tables 4.9, 4.10, 4.11, and 4.12).

(1) All C/SDR parameters have statistical significance with respect to their effect on bit loss, with SelQ being the exception (its p-value probability indicates that is not a significant factor affecting bit loss in any measure). This refutes what was found in the earlier analysis of SelQ's effect on bit loss. Data rate is also non-significant with respect to VoIP bit loss.

(2) FEC is quantitatively confirmed to have the greatest impact on bit loss in all cases.

(3) ARQ is the second most influential factor followed by TX Power.

On the following four pages are the ANOVA tables for the average effect of a parameter setting on latency. Presentation of the tables is followed by their analysis. For completeness, those parameters that are external to the C/SDR are included in the ANOVA (load, jammer power). Abbreviations in the tables are as follows J_Power (Jammer Power), SelectQ (Selective Queueing), ARQ (Automatic Repeat Request), and xmitPower (Transmit Power).

Table 4.13: ANOVA for HL Latency

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	160565864.21	56	2867247.58	10321.56	< 0.0001
A-Load	52483551.46	1	52483551.46	188931.13	< 0.0001
AF	61488801.56	3	20496267.19	73782.79	< 0.0001
F-DataRate	44813234.52	3	14937744.84	53773.13	< 0.0001
A^2	592500.72	1	592500.72	2132.89	< 0.0001
AD	167242.26	1	167242.26	602.04	< 0.0001
AH	153585.34	1	153585.34	552.88	< 0.0001
H-FEC	133181.43	1	133181.43	479.43	< 0.0001
D-ARQ	108251.21	1	108251.21	389.68	< 0.0001
DF	186733.82	3	62244.61	224.07	< 0.0001
E-FrameSize	61562.49	1	61562.49	221.61	< 0.0001
AE	51168.96	1	51168.96	184.20	< 0.0001
FH	135016.59	3	45005.53	162.01	< 0.0001
AB	20975.86	1	20975.86	75.51	< 0.0001
DE	19784.11	1	19784.11	71.22	< 0.0001
B-J_Power	17722.97	1	17722.97	63.80	< 0.0001
E^2	12427.97	1	12427.97	44.74	< 0.0001
AG	11134.47	1	11134.47	40.08	< 0.0001
EF	29566.00	3	9855.33	35.48	< 0.0001
DG	9730.63	1	9730.63	35.03	< 0.0001
G-xmitPower	9115.11	1	9115.11	32.81	< 0.0001
C-SelectQ	7071.47	1	7071.47	25.46	< 0.0001
EG	6981.30	1	6981.30	25.13	< 0.0001
AC	6690.75	1	6690.75	24.09	< 0.0001
BF	11356.10	3	3785.37	13.63	< 0.0001
CF	10256.72	3	3418.91	12.31	< 0.0001
FG	9540.21	3	3180.07	11.45	< 0.0001
G^2	2548.82	1	2548.82	9.18	0.0025
BG	1874.05	1	1874.05	6.75	0.0094
BD	1226.65	1	1226.65	4.42	0.0356
GH	1158.22	1	1158.22	4.17	0.0412
BE	449.88	1	449.88	1.62	0.2032
BH	373.19	1	373.19	1.34	0.2465
DH	271.74	1	271.74	0.98	0.3227
CD	265.67	1	265.67	0.96	0.3281
EH	183.09	1	183.09	0.66	0.4169
B^2	156.98	1	156.98	0.57	0.4522
CE	140.39	1	140.39	0.51	0.4772
CH	31.15	1	31.15	0.11	0.7377
CG	0.32	1	0.32	0.00	0.9731
BC	0.01	1	0.01	0.00	0.9956
R^2: 0.987					

Table 4.14: ANOVA for FTP Latency

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	179734056.00	56	3209536.71	10979.19	< 0.0001
A-Load	58171772.64	1	58171772.64	198994.09	< 0.0001
AF	67901843.37	3	22633947.79	77426.24	< 0.0001
F-DataRate	51833442.39	3	17277814.13	59103.97	< 0.0001
A^2	510403.47	1	510403.47	1745.99	< 0.0001
AD	178925.79	1	178925.79	612.07	< 0.0001
AH	167877.89	1	167877.89	574.28	< 0.0001
H-FEC	151757.99	1	151757.99	519.13	< 0.0001
D-ARQ	116321.30	1	116321.30	397.91	< 0.0001
E-FrameSize	80864.16	1	80864.16	276.62	< 0.0001
DF	202573.37	3	67524.46	230.99	< 0.0001
AE	64166.69	1	64166.69	219.50	< 0.0001
FH	156241.63	3	52080.54	178.16	< 0.0001
AB	23561.71	1	23561.71	80.60	< 0.0001
B-J_Power	21096.97	1	21096.97	72.17	< 0.0001
DE	19004.94	1	19004.94	65.01	< 0.0001
E^2	16349.33	1	16349.33	55.93	< 0.0001
EF	40429.73	3	13476.58	46.10	< 0.0001
AG	12311.22	1	12311.22	42.11	< 0.0001
DG	11068.77	1	11068.77	37.86	< 0.0001
G-xmitPower	10740.15	1	10740.15	36.74	< 0.0001
EG	7839.01	1	7839.01	26.82	< 0.0001
BF	14347.40	3	4782.47	16.36	< 0.0001
FG	11656.90	3	3885.63	13.29	< 0.0001
G^2	2796.13	1	2796.13	9.57	0.0020
BG	2160.04	1	2160.04	7.39	0.0066
GH	1429.73	1	1429.73	4.89	0.0270
BD	1310.01	1	1310.01	4.48	0.0343
BE	447.29	1	447.29	1.53	0.2161
BH	446.06	1	446.06	1.53	0.2168
DH	239.00	1	239.00	0.82	0.3659
B^2	219.06	1	219.06	0.75	0.3867
EH	177.00	1	177.00	0.61	0.4365
CD	62.30	1	62.30	0.21	0.6444
CH	55.77	1	55.77	0.19	0.6623
CF	83.40	3	27.80	0.10	0.9628
CE	18.24	1	18.24	0.06	0.8027
AC	13.22	1	13.22	0.05	0.8316
BC	1.64	1	1.64	0.01	0.9403
C-SelectQ	0.23	1	0.23	0.00	0.9777
CG	0.06	1	0.06	0.00	0.9890
R^2: 0.988					

Table 4.15: ANOVA for VoIP Latency

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	7998981.85	56	142838.96	596.72	< 0.0001
A-Load	1380032.51	1	1380032.51	5765.22	< 0.0001
AC	1379841.38	1	1379841.38	5764.42	< 0.0001
C-SelectQ	936775.16	1	936775.16	3913.47	< 0.0001
AF	1680931.99	3	560310.66	2340.75	< 0.0001
F-DataRate	1269403.75	3	423134.58	1767.69	< 0.0001
CF	1268567.58	3	422855.86	1766.52	< 0.0001
A^2	12521.94	1	12521.94	52.31	< 0.0001
AE	9271.05	1	9271.05	38.73	< 0.0001
AH	8335.20	1	8335.20	34.82	< 0.0001
E-FrameSize	6361.95	1	6361.95	26.58	< 0.0001
CE	6361.04	1	6361.04	26.57	< 0.0001
H-FEC	6159.03	1	6159.03	25.73	< 0.0001
CH	6157.12	1	6157.12	25.72	< 0.0001
EF	7874.62	3	2624.87	10.97	< 0.0001
FH	6881.24	3	2293.75	9.58	< 0.0001
D-ARQ	1949.60	1	1949.60	8.14	0.0043
CD	1946.36	1	1946.36	8.13	0.0044
AD	1865.23	1	1865.23	7.79	0.0053
E^2	994.03	1	994.03	4.15	0.0416
AB	763.90	1	763.90	3.19	0.0741
B-J_Power	641.43	1	641.43	2.68	0.1017
BC	639.51	1	639.51	2.67	0.1022
DF	1520.76	3	506.92	2.12	0.0957
DH	455.47	1	455.47	1.90	0.1678
AG	316.22	1	316.22	1.32	0.2504
DG	293.70	1	293.70	1.23	0.2680
G-xmitPower	286.74	1	286.74	1.20	0.2738
CG	286.03	1	286.03	1.19	0.2744
DE	189.43	1	189.43	0.79	0.3737
BF	567.74	3	189.25	0.79	0.4989
EG	139.40	1	139.40	0.58	0.4454
EH	139.23	1	139.23	0.58	0.4457
FG	296.76	3	98.92	0.41	0.7435
G^2	87.14	1	87.14	0.36	0.5463
BG	60.12	1	60.12	0.25	0.6163
GH	31.67	1	31.67	0.13	0.7161
BH	14.87	1	14.87	0.06	0.8032
BD	9.43	1	9.43	0.04	0.8426
B^2	6.75	1	6.75	0.03	0.8667
BE	4.77	1	4.77	0.02	0.8877
R^2: 0.812					

Table 4.16: ANOVA for RMAC Latency

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	1.63E-01	56	2.92E-03	6.89E+03	< 0.0001
E-FrameSize	4.64E-02	1	4.64E-02	1.09E+05	< 0.0001
F-DataRate	8.07E-02	3	2.69E-02	6.35E+04	< 0.0001
EF	2.96E-02	3	9.88E-03	2.33E+04	< 0.0001
A-Load	1.96E-03	1	1.96E-03	4.63E+03	< 0.0001
AE	1.69E-03	1	1.69E-03	3.99E+03	< 0.0001
A^2	6.13E-04	1	6.13E-04	1.45E+03	< 0.0001
AF	1.32E-03	3	4.40E-04	1.04E+03	< 0.0001
H-FEC	3.12E-04	1	3.12E-04	7.35E+02	< 0.0001
E^2	1.01E-04	1	1.01E-04	2.38E+02	< 0.0001
EH	1.01E-04	1	1.01E-04	2.38E+02	< 0.0001
D-ARQ	8.89E-05	1	8.89E-05	2.10E+02	< 0.0001
DE	7.81E-05	1	7.81E-05	1.84E+02	< 0.0001
FH	2.16E-04	3	7.19E-05	1.70E+02	< 0.0001
BD	2.66E-05	1	2.66E-05	6.28E+01	< 0.0001
B-J_Power	2.42E-05	1	2.42E-05	5.70E+01	< 0.0001
BE	2.24E-05	1	2.24E-05	5.28E+01	< 0.0001
DF	5.58E-05	3	1.86E-05	4.39E+01	< 0.0001
DH	1.64E-05	1	1.64E-05	3.88E+01	< 0.0001
EG	1.26E-05	1	1.26E-05	2.97E+01	< 0.0001
DG	1.07E-05	1	1.07E-05	2.53E+01	< 0.0001
G-xmitPower	9.11E-06	1	9.11E-06	2.15E+01	< 0.0001
AD	6.93E-06	1	6.93E-06	1.63E+01	< 0.0001
FG	1.69E-05	3	5.63E-06	1.33E+01	< 0.0001
BF	1.65E-05	3	5.50E-06	1.30E+01	< 0.0001
AB	4.32E-06	1	4.32E-06	1.02E+01	0.0014
BH	4.19E-06	1	4.19E-06	9.88E+00	0.0017
G^2	3.73E-06	1	3.73E-06	8.81E+00	0.0030
GH	2.60E-06	1	2.60E-06	6.14E+00	0.0132
AH	1.85E-06	1	1.85E-06	4.38E+00	0.0364
BG	1.38E-06	1	1.38E-06	3.25E+00	0.0714
B^2	5.63E-07	1	5.63E-07	1.33E+00	0.2489
AG	4.71E-07	1	4.71E-07	1.11E+00	0.2919
CH	1.32E-07	1	1.32E-07	3.11E-01	0.5770
AC	1.15E-07	1	1.15E-07	2.71E-01	0.6024
CD	6.79E-08	1	6.79E-08	1.60E-01	0.6890
CF	1.18E-07	3	3.95E-08	9.32E-02	0.9638
CE	3.80E-08	1	3.80E-08	8.98E-02	0.7645
C-SelectQ	3.43E-08	1	3.43E-08	8.11E-02	0.7759
CG	9.17E-09	1	9.17E-09	2.16E-02	0.8830
BC	7.36E-10	1	7.36E-10	1.74E-03	0.9668
R^2: 0.980					

The following is a summary of the major findings with respect to latency (as shown in Tables 4.13, 4.14, 4.15, and 4.16).

(1) All C/SDR parameters have statistical significance with respect to their effect on latency, with SelQ being the exception (its p-value prob<F indicates that is not a significant factor affecting latency in any measure but VoIP and HL). HL significance is due to VoIP's effect on aggregate HL performance. This correlates well with earlier assertions.

(2) Data rate is quantitatively confirmed to have the greatest impact on latency in all cases but VoIP, where SelQ is most influential.

(3) FEC is the second most influential factor followed by ARQ.

On the following four pages are the ANOVA tables for the average effect of a parameter setting on jitter. Presentation of the tables is followed by their analysis. For completeness, those parameters that are external to the C/SDR are included in the ANOVA (load, jammer power). Abbreviations in the tables are as follows J_Power (Jammer Power), SelectQ (Selective Queueing), ARQ (Automatic Repeat Request), and xmitPower (Transmit Power).

Table 4.17: ANOVA for HL Jitter

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	1.09E+01	56	1.95E-01	8.66E+00	< 0.0001
A-Load	7.57E-01	1	7.57E-01	3.36E+01	< 0.0001
AB	7.32E-01	1	7.32E-01	3.25E+01	< 0.0001
AF	1.76E+00	3	5.87E-01	2.61E+01	< 0.0001
AE	5.30E-01	1	5.30E-01	2.35E+01	< 0.0001
BE	5.20E-01	1	5.20E-01	2.31E+01	< 0.0001
F-DataRate	1.26E+00	3	4.20E-01	1.86E+01	< 0.0001
B-J_Power	4.09E-01	1	4.09E-01	1.82E+01	< 0.0001
AD	4.01E-01	1	4.01E-01	1.78E+01	< 0.0001
BF	1.18E+00	3	3.92E-01	1.74E+01	< 0.0001
BD	3.76E-01	1	3.76E-01	1.67E+01	< 0.0001
E-FrameSize	2.83E-01	1	2.83E-01	1.25E+01	0.0004
EF	8.20E-01	3	2.73E-01	1.21E+01	< 0.0001
DE	2.70E-01	1	2.70E-01	1.20E+01	0.0005
DF	6.02E-01	3	2.01E-01	8.91E+00	< 0.0001
D-ARQ	1.96E-01	1	1.96E-01	8.72E+00	0.0032
E^2	1.41E-01	1	1.41E-01	6.26E+00	0.0124
AG	7.04E-02	1	7.04E-02	3.13E+00	0.0771
BG	6.88E-02	1	6.88E-02	3.06E+00	0.0805
G^2	6.00E-02	1	6.00E-02	2.66E+00	0.1027
EG	5.73E-02	1	5.73E-02	2.54E+00	0.1107
DG	4.23E-02	1	4.23E-02	1.88E+00	0.1705
AC	3.81E-02	1	3.81E-02	1.69E+00	0.1933
BC	3.75E-02	1	3.75E-02	1.66E+00	0.1972
FG	1.04E-01	3	3.45E-02	1.53E+00	0.2037
G-xmitPower	3.28E-02	1	3.28E-02	1.46E+00	0.2273
CE	2.70E-02	1	2.70E-02	1.20E+00	0.2735
B^2	2.39E-02	1	2.39E-02	1.06E+00	0.3032
CF	6.12E-02	3	2.04E-02	9.06E-01	0.4375
C-SelectQ	1.98E-02	1	1.98E-02	8.78E-01	0.3487
CD	1.98E-02	1	1.98E-02	8.78E-01	0.3488
CH	1.69E-02	1	1.69E-02	7.52E-01	0.3857
A^2	6.66E-03	1	6.66E-03	2.96E-01	0.5865
CG	1.85E-03	1	1.85E-03	8.23E-02	0.7742
GH	7.43E-04	1	7.43E-04	3.30E-02	0.8558
AH	6.90E-04	1	6.90E-04	3.06E-02	0.8611
BH	6.44E-04	1	6.44E-04	2.86E-02	0.8657
DH	5.27E-04	1	5.27E-04	2.34E-02	0.8785
FH	7.73E-04	3	2.58E-04	1.14E-02	0.9983
H-FEC	1.45E-04	1	1.45E-04	6.43E-03	0.9361
EH	2.58E-05	1	2.58E-05	1.15E-03	0.9730
R^2: 0.06					

Table 4.18: ANOVA for FTP Jitter

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	1.09E+01	56	1.95E-01	8.66E+00	< 0.0001
A-Load	7.57E-01	1	7.57E-01	3.36E+01	< 0.0001
AB	7.32E-01	1	7.32E-01	3.25E+01	< 0.0001
AF	1.76E+00	3	5.87E-01	2.61E+01	< 0.0001
AE	5.30E-01	1	5.30E-01	2.35E+01	< 0.0001
BE	5.20E-01	1	5.20E-01	2.31E+01	< 0.0001
F-DataRate	1.26E+00	3	4.20E-01	1.86E+01	< 0.0001
B-J_Power	4.09E-01	1	4.09E-01	1.82E+01	< 0.0001
AD	4.01E-01	1	4.01E-01	1.78E+01	< 0.0001
BF	1.18E+00	3	3.92E-01	1.74E+01	< 0.0001
BD	3.76E-01	1	3.76E-01	1.67E+01	< 0.0001
E-FrameSize	2.83E-01	1	2.83E-01	1.25E+01	0.0004
EF	8.20E-01	3	2.73E-01	1.21E+01	< 0.0001
DE	2.70E-01	1	2.70E-01	1.20E+01	0.0005
DF	6.02E-01	3	2.01E-01	8.91E+00	< 0.0001
D-ARQ	1.96E-01	1	1.96E-01	8.72E+00	0.0032
E^2	1.41E-01	1	1.41E-01	6.26E+00	0.0124
AG	7.04E-02	1	7.04E-02	3.13E+00	0.0771
BG	6.88E-02	1	6.88E-02	3.06E+00	0.0805
G^2	6.00E-02	1	6.00E-02	2.66E+00	0.1027
EG	5.73E-02	1	5.73E-02	2.54E+00	0.1107
DG	4.23E-02	1	4.23E-02	1.88E+00	0.1705
AC	3.81E-02	1	3.81E-02	1.69E+00	0.1933
BC	3.75E-02	1	3.75E-02	1.66E+00	0.1972
FG	1.04E-01	3	3.45E-02	1.53E+00	0.2037
G-xmitPower	3.28E-02	1	3.28E-02	1.46E+00	0.2273
CE	2.70E-02	1	2.70E-02	1.20E+00	0.2735
B^2	2.39E-02	1	2.39E-02	1.06E+00	0.3032
CF	6.12E-02	3	2.04E-02	9.06E-01	0.4375
C-SelectQ	1.98E-02	1	1.98E-02	8.78E-01	0.3487
CD	1.98E-02	1	1.98E-02	8.78E-01	0.3488
CH	1.69E-02	1	1.69E-02	7.52E-01	0.3857
A^2	6.66E-03	1	6.66E-03	2.96E-01	0.5865
CG	1.85E-03	1	1.85E-03	8.23E-02	0.7742
GH	7.43E-04	1	7.43E-04	3.30E-02	0.8558
AH	6.90E-04	1	6.90E-04	3.06E-02	0.8611
BH	6.44E-04	1	6.44E-04	2.86E-02	0.8657
DH	5.27E-04	1	5.27E-04	2.34E-02	0.8785
FH	7.73E-04	3	2.58E-04	1.14E-02	0.9983
H-FEC	1.45E-04	1	1.45E-04	6.43E-03	0.9361
EH	2.58E-05	1	2.58E-05	1.15E-03	0.9730
R^2: 0.06					

Table 4.19: ANOVA for VoIP Jitter

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	3.94E+00	56	7.04E-02	3.87E+01	< 0.0001
A-Load	4.68E-01	1	4.68E-01	2.57E+02	< 0.0001
AC	4.40E-01	1	4.40E-01	2.42E+02	< 0.0001
C-SelectQ	2.35E-01	1	2.35E-01	1.29E+02	< 0.0001
F-DataRate	6.01E-01	3	2.00E-01	1.10E+02	< 0.0001
AF	5.10E-01	3	1.70E-01	9.36E+01	< 0.0001
AE	1.38E-01	1	1.38E-01	7.58E+01	< 0.0001
E-FrameSize	9.74E-02	1	9.74E-02	5.36E+01	< 0.0001
AD	9.28E-02	1	9.28E-02	5.10E+01	< 0.0001
CF	2.68E-01	3	8.94E-02	4.92E+01	< 0.0001
AH	8.27E-02	1	8.27E-02	4.55E+01	< 0.0001
CE	7.80E-02	1	7.80E-02	4.29E+01	< 0.0001
DE	6.91E-02	1	6.91E-02	3.80E+01	< 0.0001
EH	6.58E-02	1	6.58E-02	3.62E+01	< 0.0001
EF	1.97E-01	3	6.57E-02	3.61E+01	< 0.0001
CD	5.80E-02	1	5.80E-02	3.19E+01	< 0.0001
DF	1.55E-01	3	5.15E-02	2.84E+01	< 0.0001
D-ARQ	5.00E-02	1	5.00E-02	2.75E+01	< 0.0001
E^2	4.71E-02	1	4.71E-02	2.59E+01	< 0.0001
FH	1.40E-01	3	4.68E-02	2.57E+01	< 0.0001
DH	4.02E-02	1	4.02E-02	2.21E+01	< 0.0001
CH	3.95E-02	1	3.95E-02	2.17E+01	< 0.0001
H-FEC	2.80E-02	1	2.80E-02	1.54E+01	< 0.0001
A^2	5.66E-03	1	5.66E-03	3.11E+00	0.0778
AG	3.84E-03	1	3.84E-03	2.11E+00	0.1463
BE	3.56E-03	1	3.56E-03	1.96E+00	0.1616
BF	7.62E-03	3	2.54E-03	1.40E+00	0.2415
BD	2.47E-03	1	2.47E-03	1.36E+00	0.2436
G-xmitPower	2.43E-03	1	2.43E-03	1.34E+00	0.2479
AB	2.41E-03	1	2.41E-03	1.32E+00	0.2498
BG	2.33E-03	1	2.33E-03	1.28E+00	0.2581
CG	2.23E-03	1	2.23E-03	1.23E+00	0.2681
BH	2.09E-03	1	2.09E-03	1.15E+00	0.2840
FG	3.84E-03	3	1.28E-03	7.04E-01	0.5492
BC	1.20E-03	1	1.20E-03	6.61E-01	0.4162
GH	1.18E-03	1	1.18E-03	6.48E-01	0.4207
B-J_Power	9.75E-04	1	9.75E-04	5.36E-01	0.4639
EG	9.74E-04	1	9.74E-04	5.36E-01	0.4643
DG	6.51E-04	1	6.51E-04	3.58E-01	0.5496
B^2	1.94E-04	1	1.94E-04	1.07E-01	0.7437
G^2	8.15E-05	1	8.15E-05	4.48E-02	0.8324
R^2: 0.22					

Table 4.20: ANOVA for RMAC Jitter

Source	Sum of Squares	df	Mean Square	F Value	P Value Prob<F
Model	5.74E-04	56	1.03E-05	3.46E+01	< 0.0001
E-FrameSize	6.29E-05	1	6.29E-05	2.12E+02	< 0.0001
DE	6.29E-05	1	6.29E-05	2.12E+02	< 0.0001
D-ARQ	6.13E-05	1	6.13E-05	2.07E+02	< 0.0001
BD	2.10E-05	1	2.10E-05	7.07E+01	< 0.0001
B-J_Power	2.10E-05	1	2.10E-05	7.07E+01	< 0.0001
EF	5.90E-05	3	1.97E-05	6.63E+01	< 0.0001
BE	1.95E-05	1	1.95E-05	6.56E+01	< 0.0001
F-DataRate	5.38E-05	3	1.79E-05	6.04E+01	< 0.0001
DF	5.38E-05	3	1.79E-05	6.04E+01	< 0.0001
EG	1.79E-05	1	1.79E-05	6.04E+01	< 0.0001
DG	1.51E-05	1	1.51E-05	5.09E+01	< 0.0001
G-xmitPower	1.51E-05	1	1.51E-05	5.09E+01	< 0.0001
DH	1.01E-05	1	1.01E-05	3.40E+01	< 0.0001
H-FEC	1.01E-05	1	1.01E-05	3.40E+01	< 0.0001
EH	9.54E-06	1	9.54E-06	3.22E+01	< 0.0001
FG	2.20E-05	3	7.32E-06	2.47E+01	< 0.0001
BF	2.10E-05	3	6.99E-06	2.36E+01	< 0.0001
E^2	5.70E-06	1	5.70E-06	1.92E+01	< 0.0001
GH	4.89E-06	1	4.89E-06	1.65E+01	< 0.0001
AE	3.59E-06	1	3.59E-06	1.21E+01	0.0005
G^2	3.43E-06	1	3.43E-06	1.16E+01	0.0007
FH	7.57E-06	3	2.52E-06	8.51E+00	< 0.0001
AB	2.48E-06	1	2.48E-06	8.37E+00	0.0038
A-Load	2.26E-06	1	2.26E-06	7.63E+00	0.0057
AD	2.26E-06	1	2.26E-06	7.63E+00	0.0057
A^2	1.45E-06	1	1.45E-06	4.89E+00	0.0271
B^2	1.01E-06	1	1.01E-06	3.42E+00	0.0644
AH	9.88E-07	1	9.88E-07	3.33E+00	0.0680
BH	7.98E-07	1	7.98E-07	2.69E+00	0.1011
CH	2.16E-07	1	2.16E-07	7.27E-01	0.3939
AF	6.46E-07	3	2.15E-07	7.26E-01	0.5362
CF	5.68E-07	3	1.89E-07	6.38E-01	0.5904
CE	1.59E-07	1	1.59E-07	5.37E-01	0.4639
AG	4.10E-08	1	4.10E-08	1.38E-01	0.7099
BC	1.25E-08	1	1.25E-08	4.20E-02	0.8376
C-SelectQ	1.19E-08	1	1.19E-08	4.01E-02	0.8412
CD	1.19E-08	1	1.19E-08	4.01E-02	0.8412
AC	1.94E-09	1	1.94E-09	6.55E-03	0.9355
BG	1.03E-09	1	1.03E-09	3.49E-03	0.9529
CG	1.07E-11	1	1.07E-11	3.62E-05	0.9952
R^2: 0.20					

The following is a summary of the major findings with respect to jitter (as shown in Tables 4.17, 4.18, 4.19, and 4.20).

(1) Data Rate, FrameSize and ARQ were statistically significant with respect to their effect on jitter. However, for HL and FTP, xmitPower, SelQ and FEC have no statistical significance. Additionally, xmitPower was not significant for VoIP jitter. SelQ had no statistical effect on any metric but VoIP jitter (correlating well with earlier assertions).

(2) Data rate is quantitatively confirmed to have the greatest impact on jitter in HL and FTP responses. However, for VoIP SelQ is most influential, and for RMAC FrameSize was most influential (findings that track well with our earlier analysis).

(3) FrameSize was the second most influential factor impacting jitter followed by ARQ.

4.1.2.1 Analysis of Variance - Summary of Findings

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm.

- **Throughput** - The ANOVA confirms observations made during the analysis of the average effect of a parameter setting. Data rate is the most significant parameter effecting throughput (with the notable exception of SelQ being the most effective on VoIP throughput). FEC is next followed by xmitPower/ARQ, with ARQ have a slightly higher impact.

- **Bit Loss** - The ANOVA for bit loss nearly confirms our earlier observations. FEC is the single greatest parameter affecting bit loss. However, ARQ is quantitatively shown to be the second followed by xmitPower (analysis of the average effect of a parameter setting had these reversed). Also, SelQ does not have a statistically significant effect on bit loss.

- **Latency** - The ANOVA for latency confirms our earlier observations. Data rate is again the most significant factor impacting latency (with the exception of SelQ being the primary factor impacting VoIP latency). This is followed by FEC and then ARQ.

- **Jitter** - The most significant factors are data rate followed by FrameSize and then ARQ (SelQ is the most significant influence on VoIP jitter). These results confirm earlier observations.

Table 4.21: Top Three Factors Influencing Response

	Throughput	Bit Loss	Latency	Jitter
1	DataRate	FEC	DataRate	DataRate
2	FEC	ARQ	FEC	FrameSize
3	ARQ	xmitPower	ARQ	ARQ

Table 4.21 summarizes the top three parameters impacting each metric. While an ANOVA allows one to quantitatively determine which factors (or their interactions) most impact a response, it does not indicate whether the impact is positive or negative. In order to determine the effect, graphs for multi-factor interactions are analyzed. The ANOVA tables in combination with identification of the top three factors influencing each metric guide our analysis of the multi-factor interactions. Those factors in Table 4.21 are combined to determine whether a factor's effect is positive or negative, or if there is an interaction that was not evident in the earlier analysis.

4.1.3 Impact of a Parameter and Multi-Factor Analysis

This subsection presents the multi-factor analysis. This phase of the research is targeted at identifying the effect (good or bad) of the statistically significant parameters on a response. Also, this analysis will uncover any multi-factor interactions. For example, one may find that at the lower data rates it is advantageous to enable FEC while at the higher data rates it is not.

Before preceding with the multi-factor analysis, it is worth noting that R^2, a measure of how well a regression line approximates the real data points, for the various predictive models associated with the DOE analysis varies from metric to metric, with throughput and latency models providing a nearly perfect fit (R^2 >0.98). In other words, based on the data that was collected the relative predictive power of these models is very good. R^2 for bit loss is a somewhat less respectable at 0.59, but one can still make the case that it is correct most of the time. However, R^2 for jitter is an abysmal 0.06, confirming what was had seen in the analysis of jitter with the average effect of a parameter setting. Because the multi-factor analysis relies on the predictive model to calculate factor interactions, one cannot justify using this method to draw any conclusions about jitter.

The following three pages present the multi-factor interactions for throughput. The charts are discussed in order of significance based on the ANOVA from Table 4.21. The chart is organized as follows, the Y-axis is the response, the X1-axis is factor one, and the X2-axis is factor two. The presentation order is as follows, the interaction of Data Rate and FEC, followed by Data Rate and ARQ, and then by FEC and ARQ.

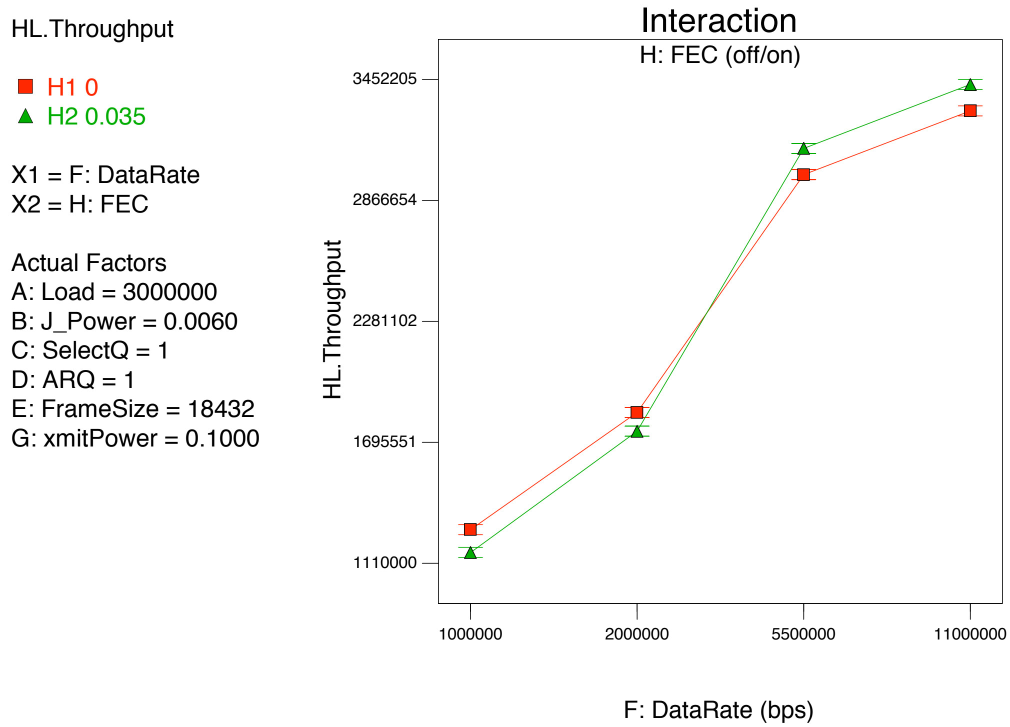

Figure 4.17: Interaction of Data Rate and FEC on Throughput

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of Data Rate and FEC (as shown in Figure 4.17).

- In all cases, increasing data rate positively impacted throughput.
- A data rate increase when combined with enabling of FEC improved throughput with the following exception. There was a negative impact on throughput when transmitting 1 or 2 Mbps with medium or high power, and ARQ is enabled.

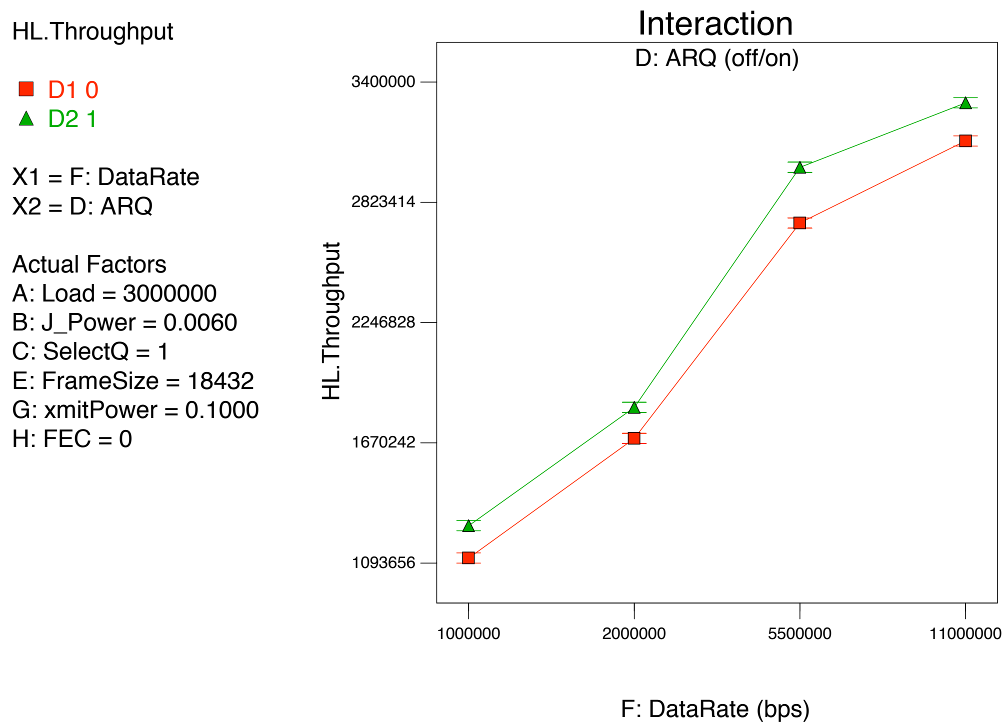

Figure 4.18: Interaction of Data Rate and ARQ on Throughput

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of Data Rate and ARQ (as shown in Figure 4.18).

- This chart illustrates the general trend for enabling of ARQ on throughput (with the noted exception discussed earlier). In all remaining cases ARQ has a positive impact.

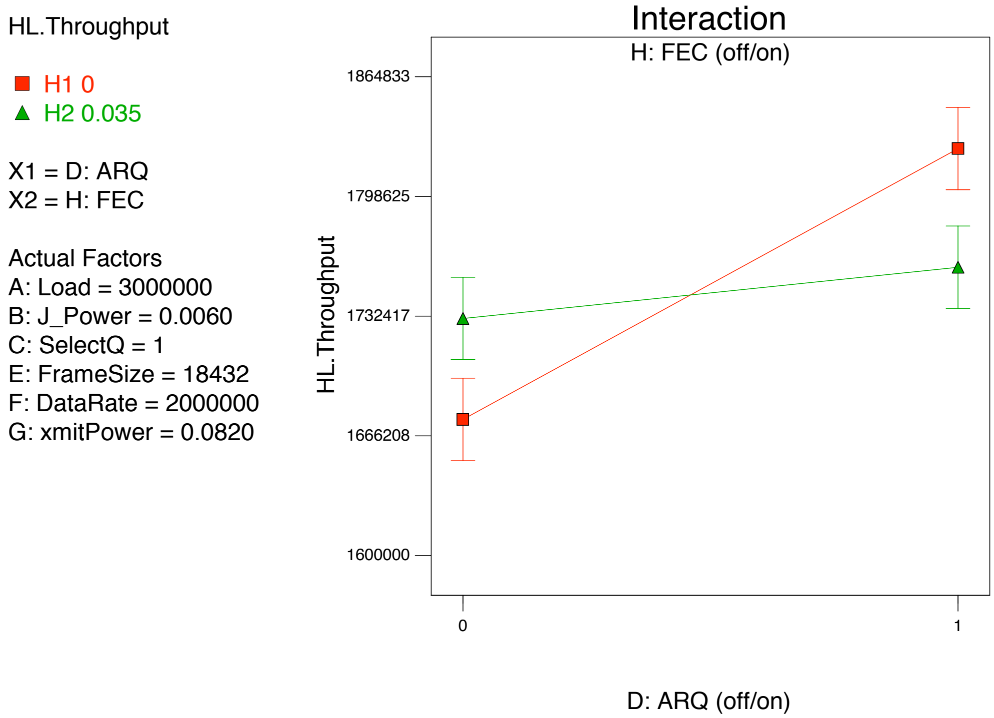

Figure 4.19: Interaction of FEC and ARQ on Throughput

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of FEC and ARQ (as shown in Figure 4.19).

- This chart confirms earlier analysis. Enabling both FEC and ARQ at the lower data rates has negative impact on throughput. In all other cases the interaction was positive.

The following three pages present the multi-factor interactions for bit loss. The charts are discussed in order of significance based on the ANOVA from Table 4.21. The chart is organized as follows, the Y-axis is the response, the X1-axis is factor one, and the X2-axis is factor two. The presentation order is as follows, the interaction of FEC and ARQ, followed by FEC and xmitPower, and then by ARQ and xmitPower.

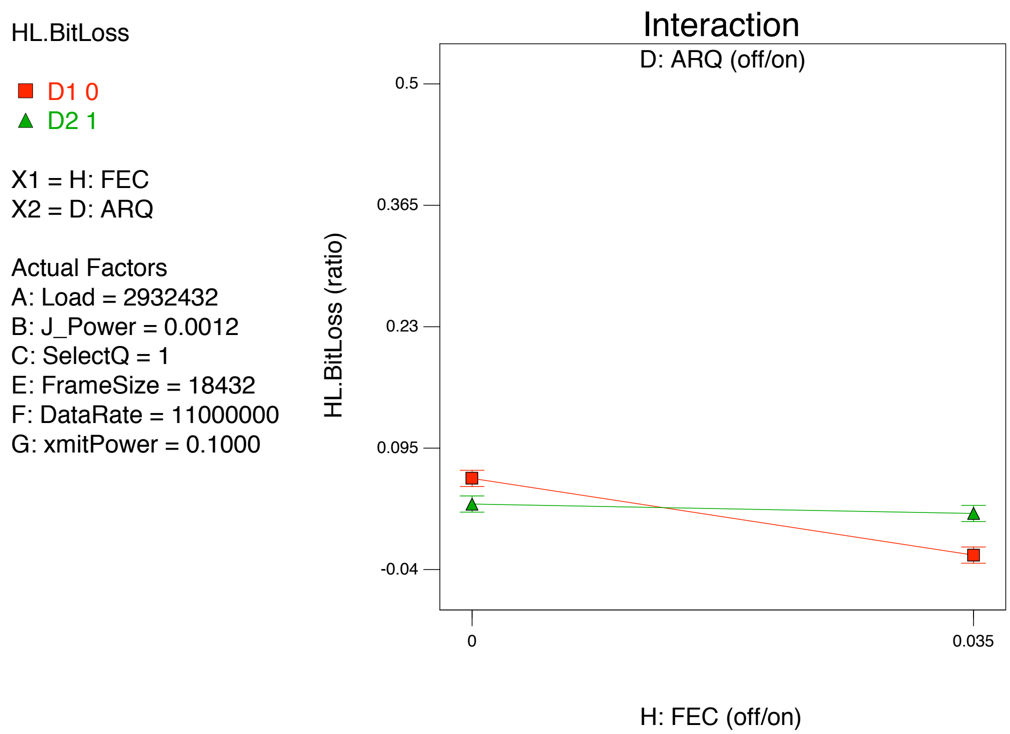

Figure 4.20: Interaction of FEC and ARQ on Bit Loss

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of FEC and ARQ (as shown in Figure 4.20).

- The enabling of FEC and ARQ have a positive effect on bit loss, with the exception of the 11 Mbps data rate.

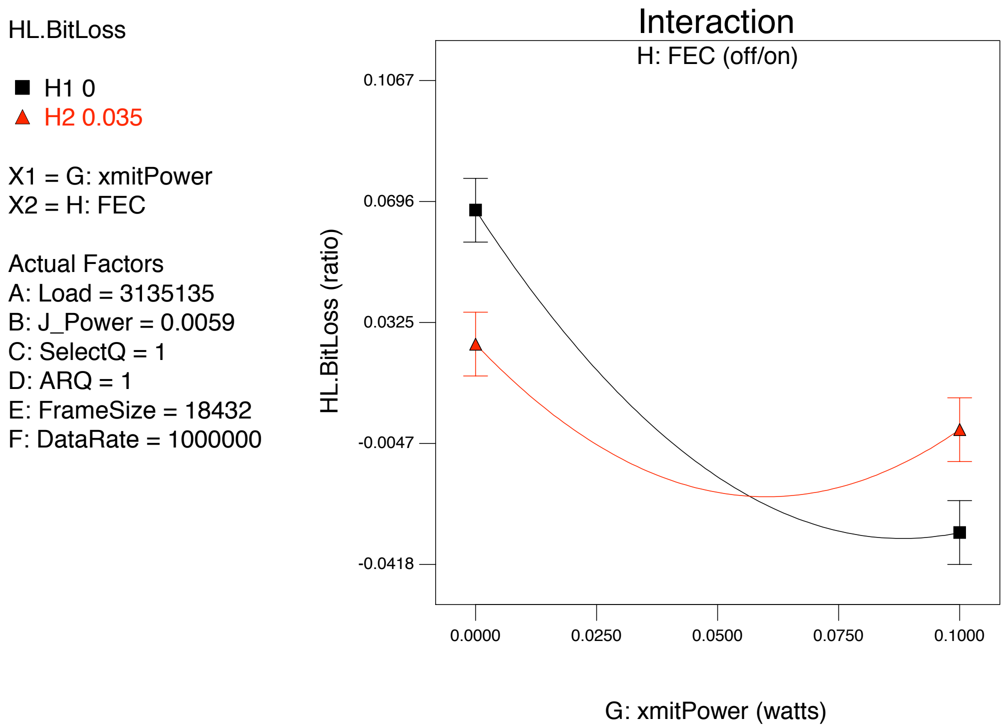

Figure 4.21: Interaction of FEC and xmitPower on Bit Loss

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of FEC and xmitPower (as shown in Figure 4.21).

- An increase in xmitPower with FEC enabled has beneficial effects in all cases but one, at the 1 Mbps data rate enabling of FEC and increasing xmitPower has a negative effect.

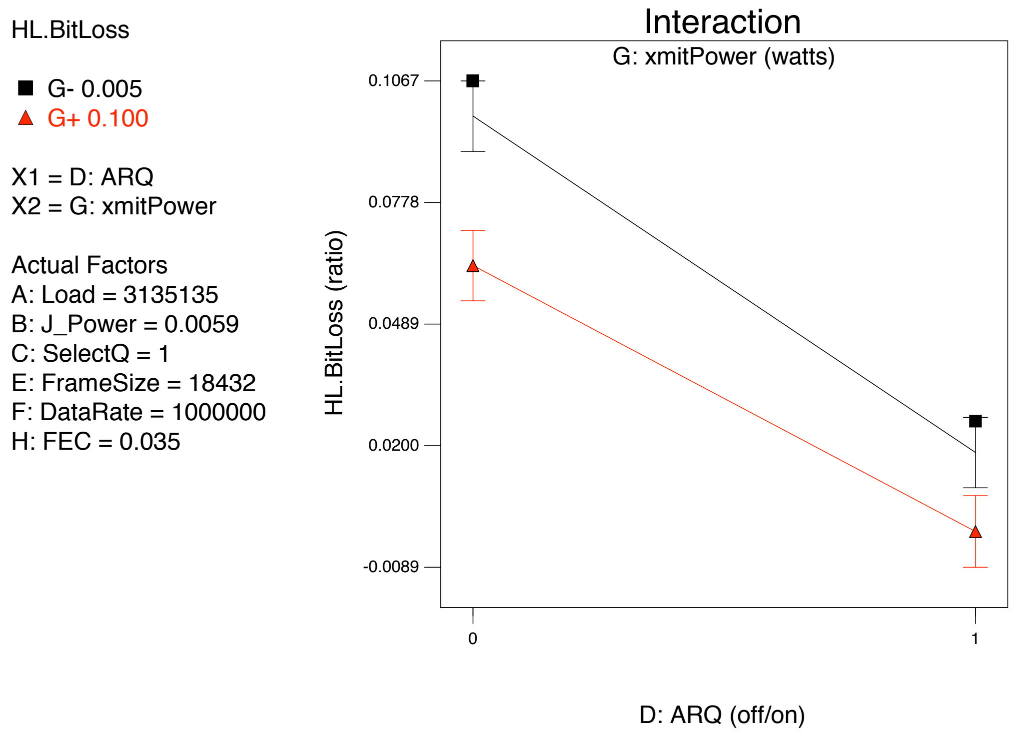

Figure 4.22: Interaction of ARQ and xmitPower on Bit Loss

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of ARQ and xmitPower (as shown in Figure 4.22).

- ARQ and xmitPower interact in a positive way with the exception of the negative impact at 11 Mbps when FEC and ARQ are enabled (as discussed in the analysis of FEC and ARQ).

The following three pages present the multi-factor interactions for latency. The charts are discussed in order of significance based on the ANOVA from Table 4.21. The chart is organized as follows, the Y-axis is the response, the X1-axis is factor one, and the X2-axis is factor two. The presentation order is as follows, the interaction of Data Rate and FEC, followed by Data Rate and ARQ, and then by FEC and ARQ.

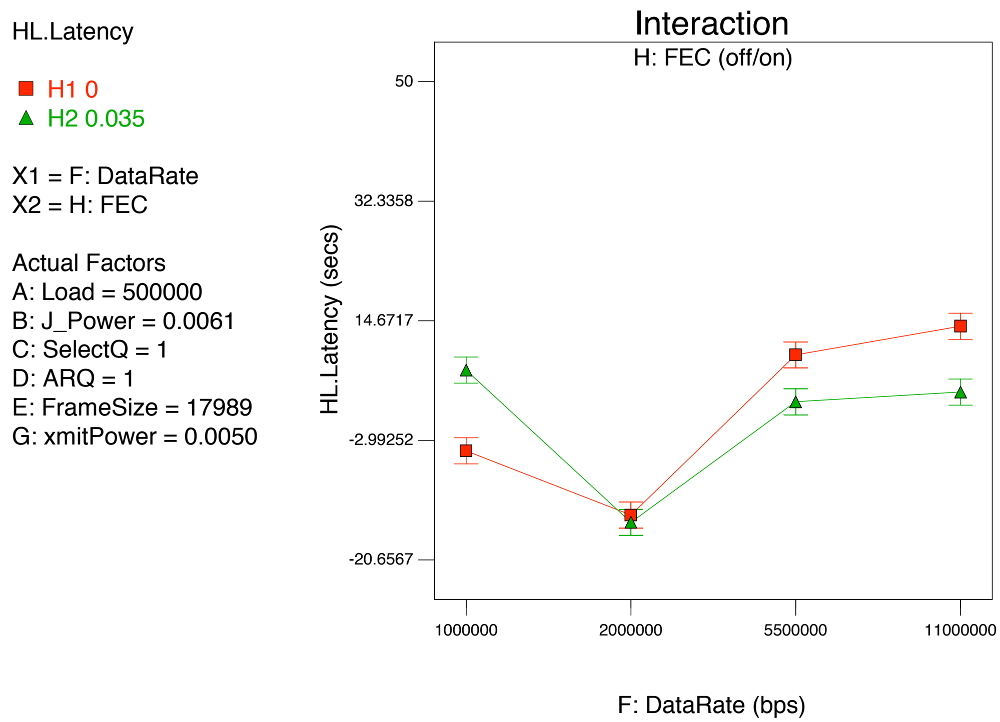

Figure 4.23: Interaction of Data Rate and FEC on Latency

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of Data Rate and FEC (as shown in Figure 4.23).

- Enabling FEC in combination with Data Rate has a harmful effect in all but the following, when load on the system is at 0.5 Mbps and Data Rate is at either 5 or 11 Mbps, enabling FEC decreases latency.

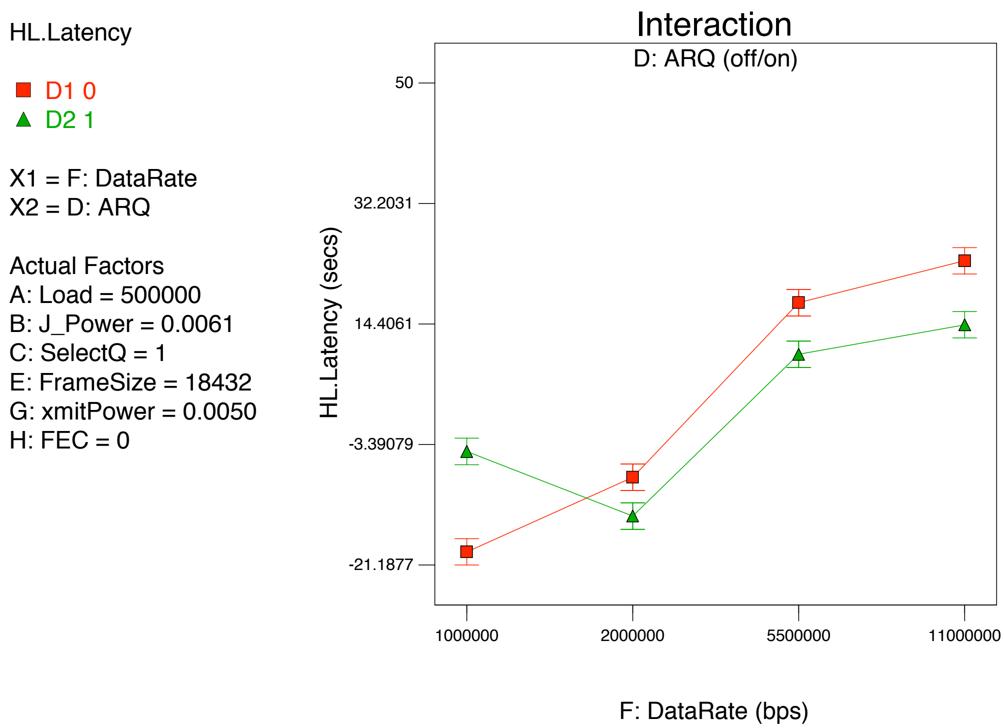

Figure 4.24: Interaction of Data Rate and ARQ on Latency

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of Data Rate and ARQ (as shown in Figure 4.24).

- Typically ARQ in combination with Data Rate has a harmful effect, however, there were some interesting exceptions.
  - ∗ At medium or high xmitPower and under either medium or high load, enabling ARQ had a slight positive effect.
  - ∗ When the system is under low or medium load at the 1 Mbps data rate, enabling ARQ has a positive effect.

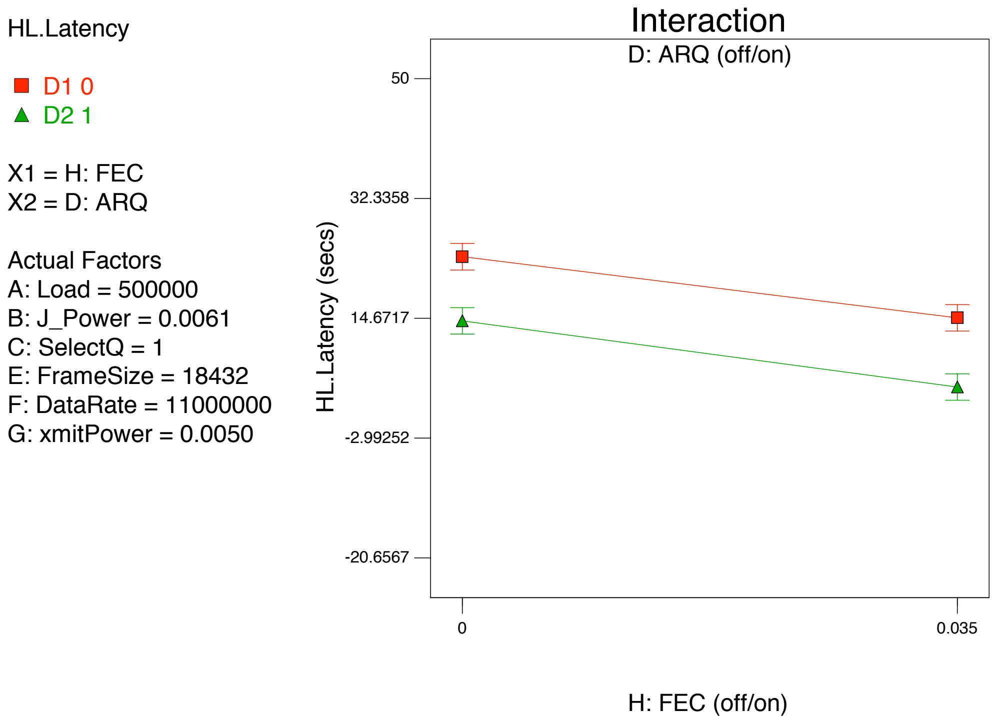

Figure 4.25: Interaction of FEC and ARQ on Latency

The following is a list of the findings that are most relevant to the development of the reconfiguration algorithm with respect to the interaction of FEC and ARQ (as shown in Figure 4.25).

- Generally, the combination of FEC with ARQ has a negative impact on latency with the following exception. Recall from the analysis of Data Rate and FEC, that when the system is under low load the combination of FEC and ARQ at the 5 and 11 Mbps data rates improved latency.

As discussed earlier, the predictive model for jitter is very suspect, with an R^2 of 0.06. Upon inspection of the multi-factor interaction charts for jitter, one cannot report any trends with confidence, other than that the predictive model is very poor. Figure 4.26 was typical of the factor interactions with respect to jitter and yielded no additional information.

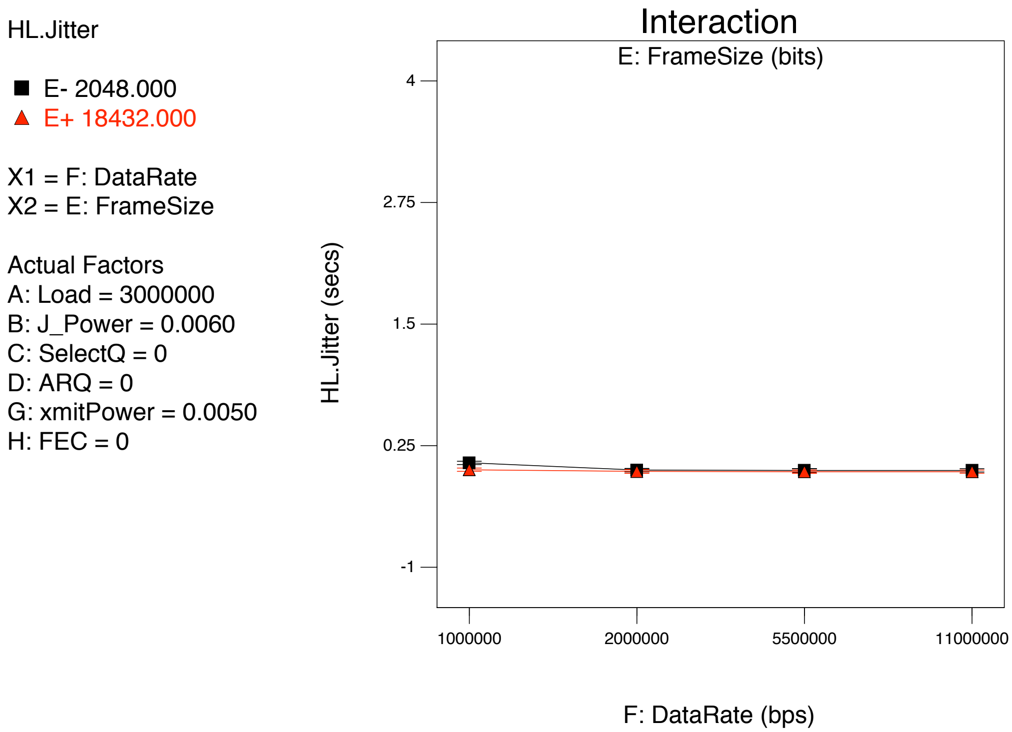

Figure 4.26: Interaction of Data Rate and FrameSize on Jitter

4.1.3.1 Multi-Factor Analysis - Summary of Findings

Table 4.22: Top Three Factors Influencing Response and Their Effect

	Throughput	Effect	Bit Loss	Effect	Latency	Effect
1	DataRate	+	FEC	+	DataRate	+
2	FEC	+	ARQ	+	FEC	-
3	ARQ	+	xmitPower	+	ARQ	-

Table 4.22 is summary of the effects a parameter setting had on the response. However, there are some caveats to the table which were exposed in the multi-factor analysis. They are summarized below.

- **Throughput** - ARQ and FEC enabled at 1 and 2 Mbps data rates had a negative impact on throughput.

- **Bit Loss** - FEC and ARQ enabled at the 11 Mbps data rate had a negative impact on bit loss. Also, at the 1 Mbps data rate increasing power in combination with enabling FEC resulted in higher bit loss.

- **Latency** - When transmitting at 5.5 or 11 Mbps in combination with a load of 0.5 Mbps, enabling FEC decreased latency.

While the ANOVA allowed determination of which factors most impact a response, it does not indicate whether the impact is positive or negative. This phase of the research focused on discovering whether a parameter positively or negatively affected performance, and uncovered any multi-factor interactions. As confirmed in this analysis, and shown in Table 4.22, there is now a experimentally supported and quantitatively confirmed basis for development of a tuning strategy for a C/SDR. For example, one can say with some certainty (minus caveats) that to improve throughput one can begin by increasing data rate, and then enabling FEC, and if performance goals are still not being met, enable ARQ.

4.1.4 Experimental Analysis - Summary of Findings

This work's experimental analysis has progressed from the average effect of a parameter (shows the magnitude of effect a parameter has on the system in general), to an ANOVA (determines which and orders parameters having the greatest impact), and finally the multi-factor analysis (determines whether changing a parameter is good or bad and uncovers multi-factor interactions). In other words, the first phase of research set out to uncover which "knobs" to turn, whether or not to turn them, what order to turn them in, and any unforeseen side-effects of turning the knobs. Table 4.22 is the culmination of that effort. This table drives the development of the algorithm presented in the second phase of the research.

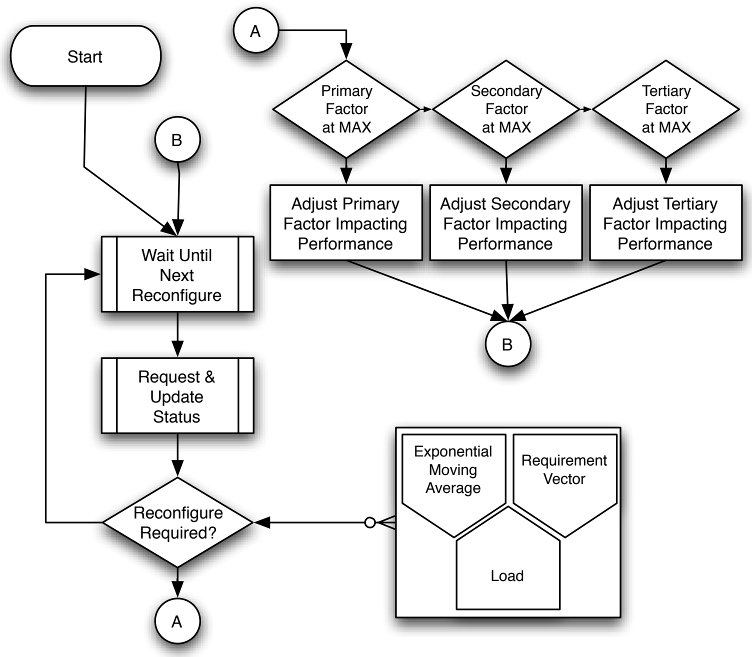

Figure 4.27: Reconfiguration Algorithm

4.2 Algorithm Performance

The algorithm for controlling reconfiguration of the system was designed to take advantage of the experimental analysis. The thesis of this work is that a C/SDR can improve wireless performance by exploiting cross-layer parametric optimization in the presence of an active source of noise. Achieving this thesis is the primary goal of the algorithm for determining the next configuration of the radio (see Section 3.4 for a detailed description of how the algorithm operates). This decision is based on performance information (throughput, bit loss, latency) and the requirements vector (goals for throughput, bit loss and latency). If the C/SDR is performing in accordance with the goals in the requirements vector then a configuration change to improve performance is not required. An Exponential Moving Average (EMA) is used to track system performance. However, there are secondary concerns as well. It makes sense when considering resource use and fairness to operate in a configuration that meets requirements, yet minimizes power output as well as time on the link. By minimizing the time on the link and using the lowest possible power settings, a pair of nodes reduces the chance of interfering with other communicating nodes (or being detected by an emissions seeking weapon). Additionally, the reconfiguration algorithm starts with the most conservative initial settings and becomes more aggressive as needed. Hereafter the reconfiguration algorithm is referred to as, Most Conservative Configuration (MCC). The algorithm, MCC, uses a **greedy** approach to resolving performance problems. When the system is not meeting a performance goal it makes use of Table 4.22 to determine which factors to adjust and in what order. For example, if the EMA for bit loss is too high, the algorithm would start with the factor which most positively impacts bit loss (FEC) and enable it. This process is repeated, through secondary and tertiary factors, until the system either meets the performance goal or it is unable to change configuration to effect a positive change (see Figure 4.27).

4.2.1 Comparative Performance

Table 4.23: Best and Worst Case Static Configurations

	SelQ	ARQ	FrameSize	DataRate	xmitPower	FEC
Best	on	on	largest	11 Mbps	max	on
Worst	off	off	smallest	1 Mbps	lowest	off

On the following pages are a set of charts which compare the performance of MCC with two static configurations of the C/SDR, the best and worst case settings. The settings were derived from the experimental performance data and are shown in Table 4.23. MCC starts with the most conservative settings, which are equivalent to the worst case settings, and then adapts to meet performance goals. The following pages compare performance of the best and worst case static configurations with MCC in respect to throughput, bit loss, latency, jitter and power output.

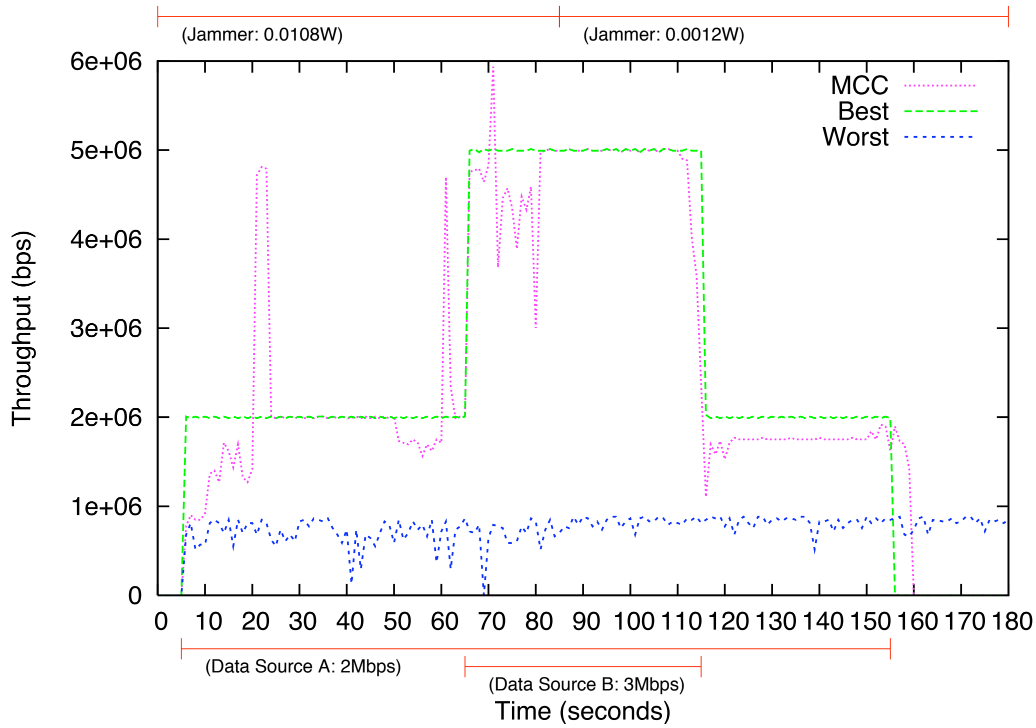

Figure 4.28: MCC vs. Best and Worst Static Configurations on Throughput

Figure 4.28 shows the best and worst case static configurations compared to the reconfiguration algorithm with respect to throughput. The scenario is as follows (and is repeated for bit loss, latency, and jitter). The duration of the run is 180 seconds. The jammer is transmitting at 0.0108 Watts from 0 to 85 seconds, at which time it decreases transmit power to 0.0012 Watts for the remainder of the experiment (as shown in a red line above the chart). Additionally, there are two data sources, source A is transmitting data at 2 Mbps from 5 to 155 seconds and source B is transmitting at 3 Mbps from 65 to 115 seconds (as shown in red lines below the chart). Therefore, the radio must contend with changes in the environment as well as in system load. The requirements vector given to MCC stipulates an initial throughput goal of 2 Mbps, which is increased to 5 Mbps when the second load is applied to the system, and then drops back to 2 Mbps in 115th second when the additional load stops. The second component of the requirements vector requires the MCC algorithm to maintain a bit loss of less than

5%. Additionally, the reconfiguration interval in these experiments is set for 10 seconds (i.e., there is a statistics exchange and potential reconfiguration every 10 seconds). The following list summarizes the major observations for this chart.

- The best case and MCC clearly outperform the worst case configuration. The best case configuration delivering all data in the 156th second of the experiment, followed by MCC in the 159th second (+2%), and trailed by the worst case which finishes delivery of the data in the 544th second (3.5x).
- The drops in throughput are due to noise bursts from the jamming node. The best case static configuration delivers the data without a detectible drop in throughput, whereas MCC and worst case are effected to a greater degree.
- MCC also shows some spikes in throughput. These spikes are attributed to data queued at the MAC layer and subsequent reconfiguration to a higher data rate (thus the backlog of frames are dumped onto the media).

The following page presents the results for bit loss.

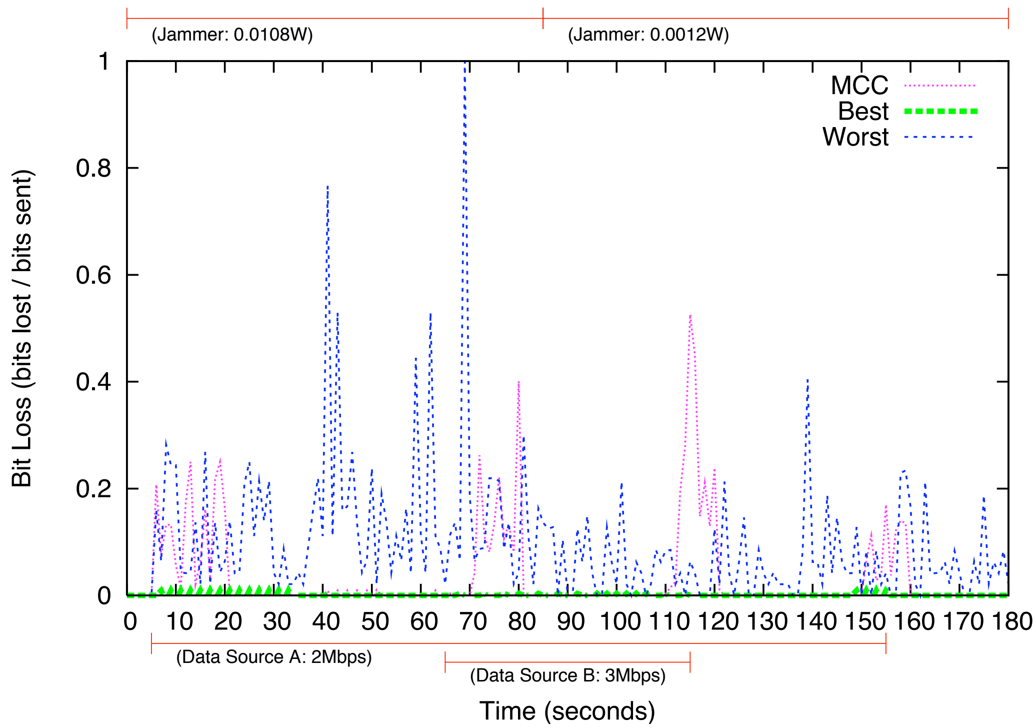

Figure 4.29: MCC vs. Best and Worst Static Configurations on Bit Loss

Figure 4.29 shows the best and worst case static configurations compared to the reconfiguration algorithm (MCC) with respect to bit loss (note, the scenario remains the same as the previous run). The following list summarizes the major observations for this chart.

- The best case and MCC clearly outperform the worst case configuration. The best case configuration delivers all data with a 0% bit loss, followed by MCC with a slightly higher percentage, 3.67%, and trailed by the worst case which is considerably poorer, 9.89%.
- It is also important to note that the MCC algorithm is being driven by a bit loss goal of 5%, which it maintains on average.
- Bit loss is directly attributed to noise bursts from the jamming node.

The following page presents the results for latency.

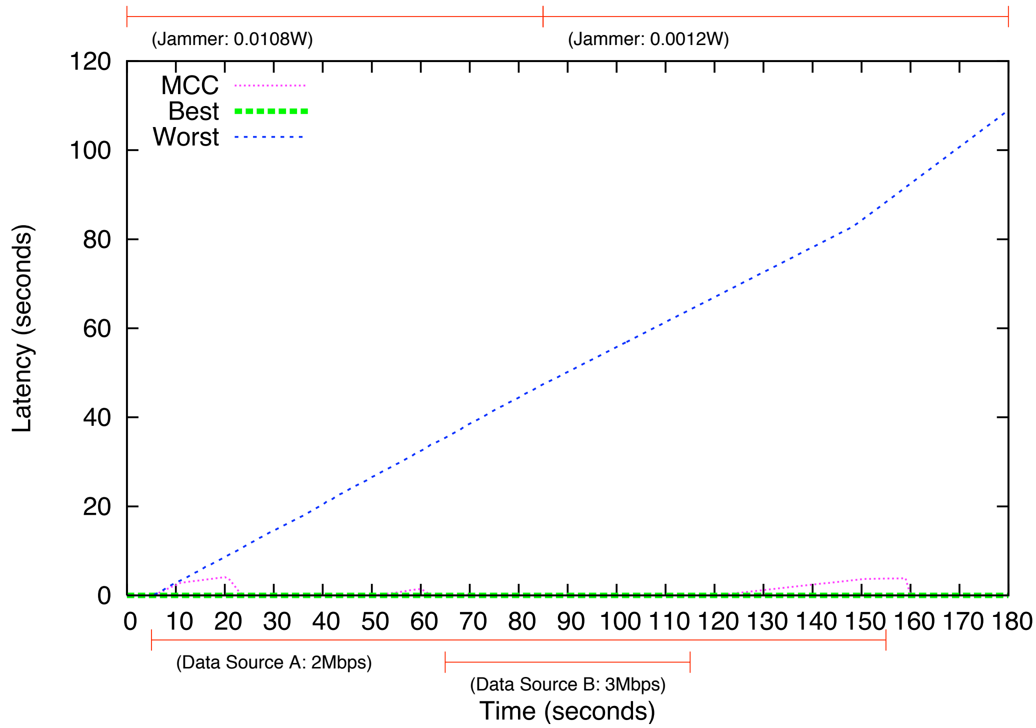

Figure 4.30: MCC vs. Best and Worst Static Configurations on Latency

Figure 4.30 shows the best and worst case static configurations compared to the reconfiguration algorithm (MCC) with respect to latency. The following list summarizes the major observations for this chart.

- The performance of MCC closely mirrors that of the best configuration, with the exception of those periods when the reconfiguration algorithm lags demand on the system. This lag is due to the use of the exponential moving average (EMA). EMA's smoothing provides balance and prevents the system from oscillating between configurations. The average latency for the best configuration is 0.017 seconds, followed by MCC at 0.8412 seconds. MCC is higher due to time required to reconfigure and preference of meeting bit loss goals over latency.

- The worst configuration is unable to deliver the data during the run due to queuing at the MAC layer, its latency average for the run is 50.28 seconds.

The following page presents the results for jitter.

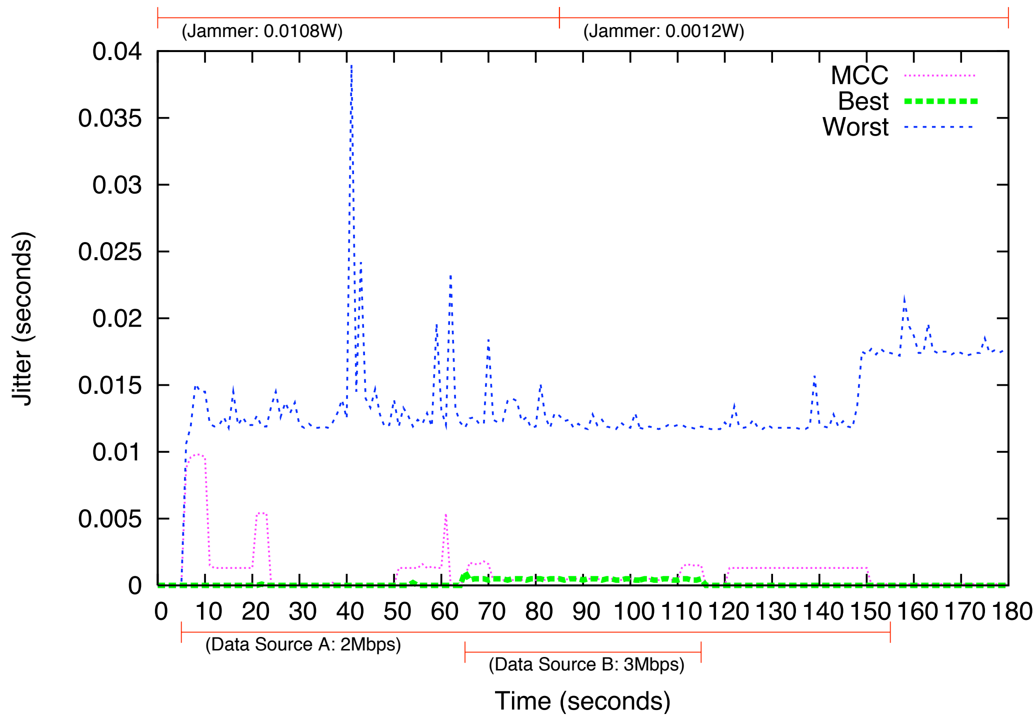

Figure 4.31: MCC vs. Best and Worst Static Configurations on Jitter

Figure 4.31 shows the best and worst case static configurations compared to the reconfiguration algorithm (MCC) with respect to jitter. The following list summarizes the major observations for this chart.

- The performance of MCC closely mirrors that of the best configuration, with the exception of those periods when the reconfiguration algorithm is impacted by changing configuration and noise. The best configuration has a jitter of 0.0001 seconds, and MCC has a jitter of 0.0009 seconds.

- The worst configuration is unable to deliver the stability of the other methods, performing at 0.0133 seconds.

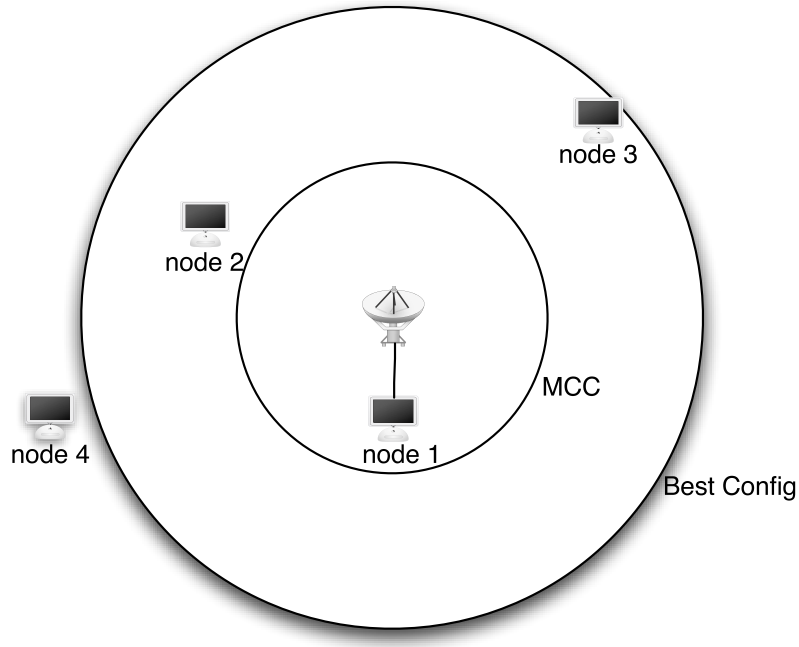

Figure 4.32: Radio Frequency Propagation for MCC vs Best Configuration

4.2.1.1 Summary of Findings

In nearly all cases MCC is a close second with respect to metrics measured. In interpreting these results it is important to realize that although the best case static configuration outperforms MCC it does not do so by a significant margin. Additionally, MCC has an average power output of 0.0236 watts, while the best case uses 0.1 watts, 76% less efficient. This difference in power output translates into energy propagation as shown in Figure 4.32. In this scenario, the center node is communicating with node 1. MCC is much more power efficient, having less impact on neighboring nodes, whereas the best configuration could potentially disrupt communication in nodes 1, 2 and 3. The following two pages summarize the average performance of the three methods with respect to throughput, bit loss, latency, and jitter.

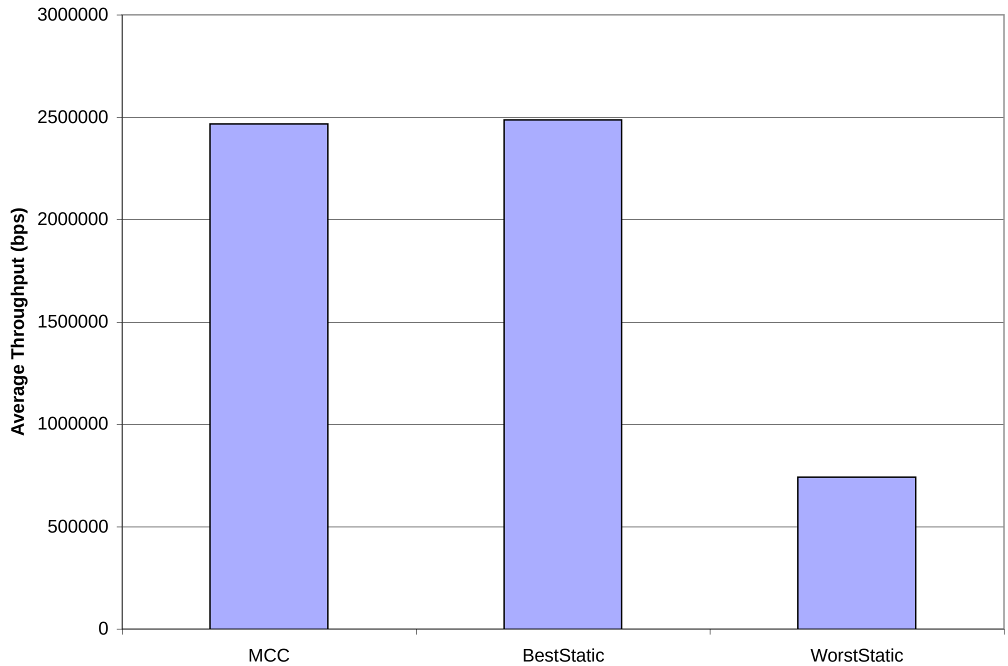

Figure 4.33: MCC vs. Best and Worst Static Configurations - Average Throughput

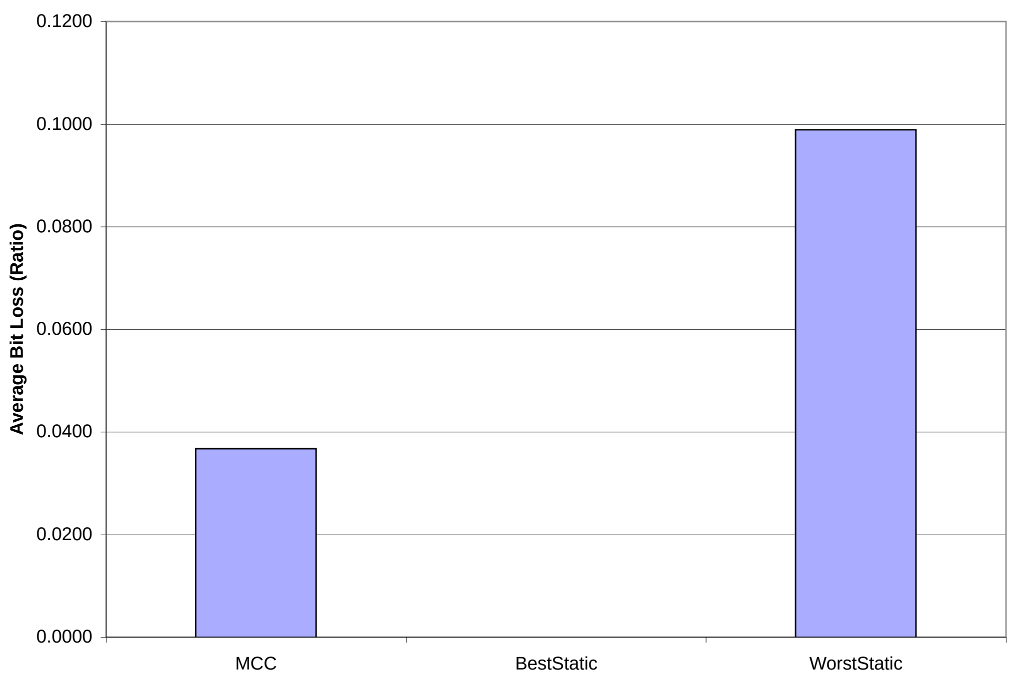

Figure 4.34: MCC vs. Best and Worst Static Configurations - Average Bit Loss

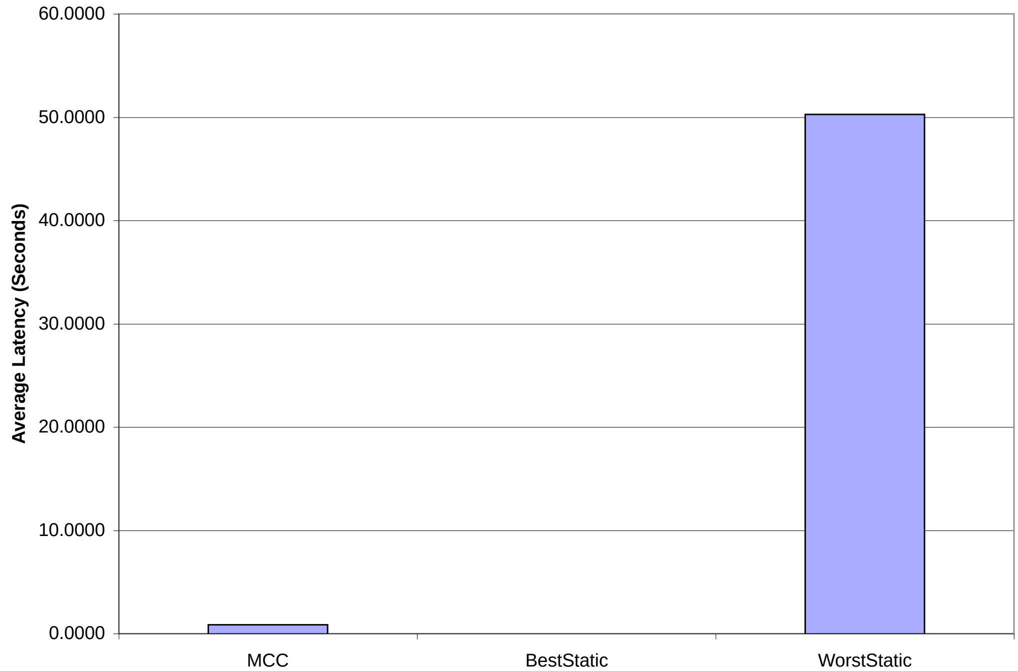

Figure 4.35: MCC vs. Best and Worst Static Configurations - Average Latency

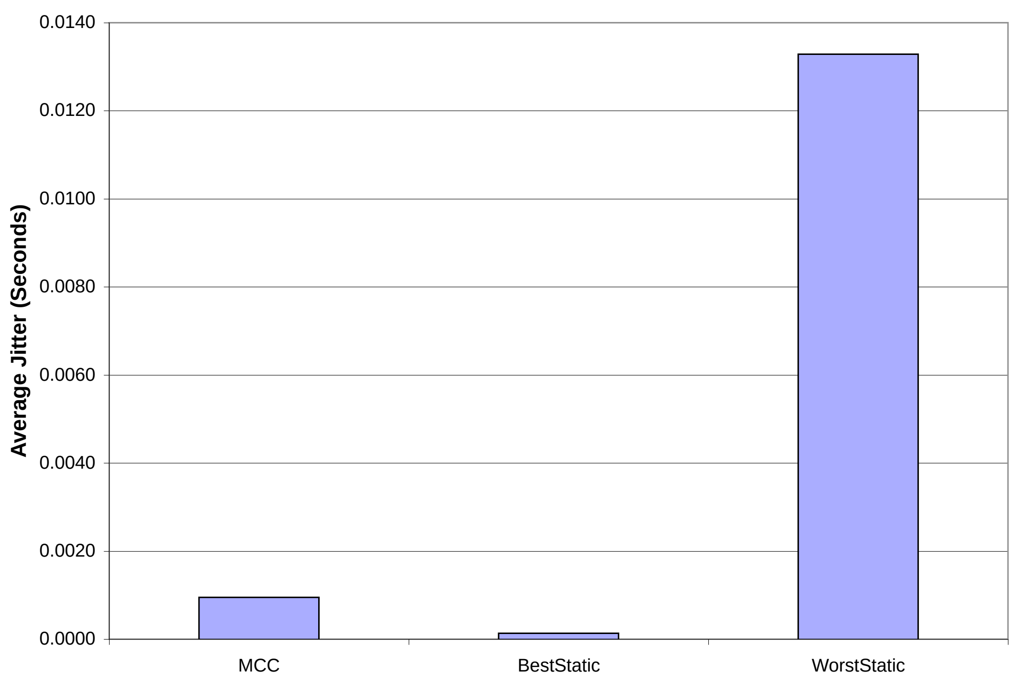

Figure 4.36: MCC vs. Best and Worst Static Configurations - Average Jitter

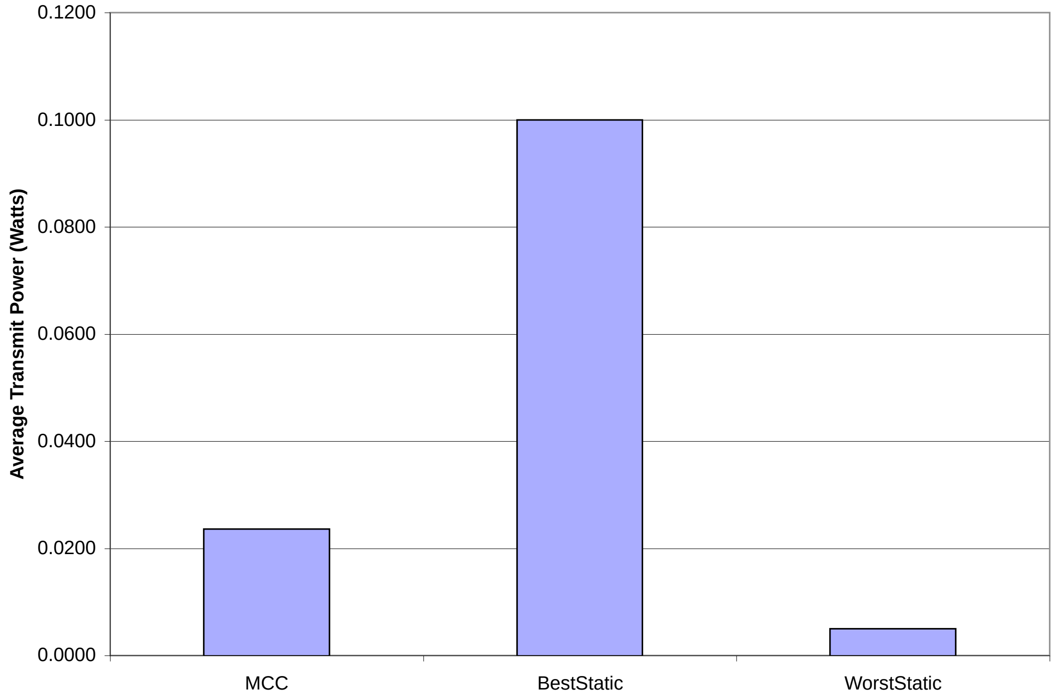

Figure 4.37: MCC vs. Best and Worst Static Configurations - Average Power Output

As shown in Figure 4.37, MCC clearly outperforms the Best static configuration and closely matches it with respect to latency, jitter, and bit loss.

4.2.2 General Performance

This section reports on the reconfiguration algorithm (e.g., time to reconfigure). This is a measure of how long it takes the C/SDR to change from one set of parametric settings to another. This metric is averaged across each run and reported for each reconfiguration interval. The following chart summarizes the time to configure for the MCC algorithm. This is a measure of how long it takes the C/SDR to change from one set of parametric settings to another. This metric is reported for each reconfiguration interval (see Figure 4.38). The average time to reconfigure is 0.0015 seconds. Variances in reconfiguration times are caused by interference due to the jamming node.

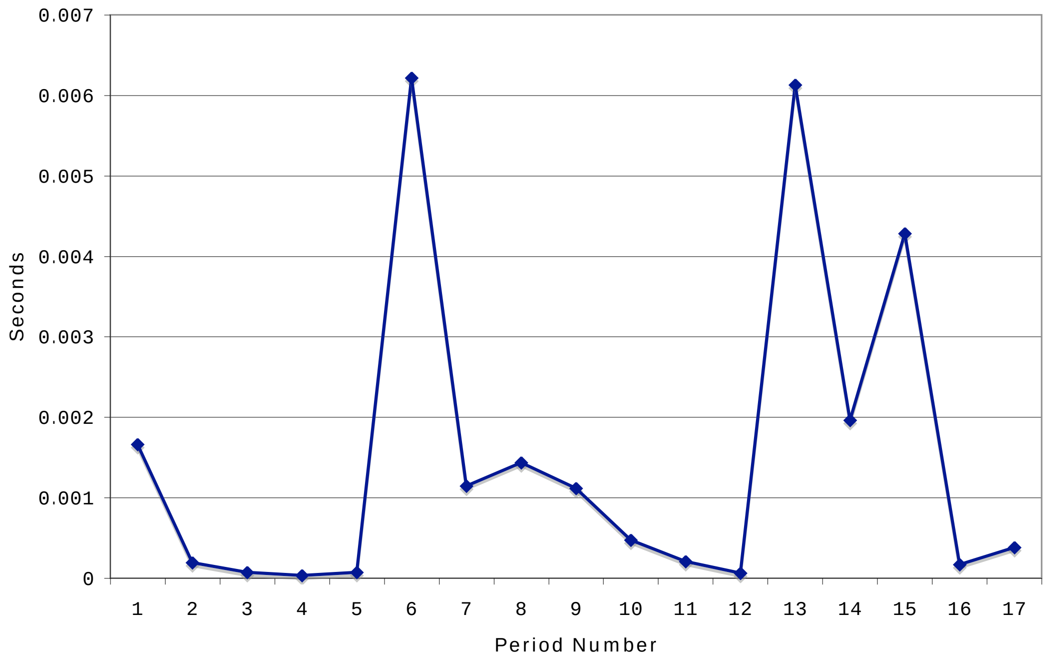

Figure 4.38: Time to Reconfigure

4.2.3 Algorithm - Summary of Findings

The MCC algorithm offers a sound approach to dynamically reconfiguring a C/SDR in the presence of noise. MCC closely matches the performance of a statically configured radio and surpasses it with respect to efficiency in transmit power. Additionally, the MCC algorithm is able dynamically to adapt to changes in load and bit loss while maintaining performance goals. Also, the MCC algorithm does not require a priori knowledge. It can maintain the most conservative configuration while meeting performance goals, while the static configurations are only as good as their initial settings. Also, there is a significant range in performance from the best and worst case settings, therefore there is a large potential for variance in the static configuration's performance based upon how well one chooses the initial configuration. The next chapter will summarize the major findings, contributions, and future research directions associated with this work.

Chapter 5

Conclusion

A C/SDR system that is experiencing active interference can improve performance by exploiting dynamic cross-layer parametric optimization.

Above is the thesis of this work, which has been proven through experimental analysis (DOE and ANOVA techniques) and validated through the development of the Most Conservative Configuration (MCC) algorithm. The following sections detail the major contributions, findings, and future work related to this research.

5.1 Contributions

The first major contribution of this work was the experimental parametric analysis of the C/SDR platform. By using Design of Experiments (DOE) and Analysis of Variance (ANOVA) one is able to determine which settings most affect a C/SDR's performance and how the settings interact. The second major contribution of this research was the development and validation of the MCC algorithm. This algorithm used the experimental analysis as a basis for achieving desired performance goals while maintaining the most conservative configuration. The next major contribution of this work was the development, testing, and evaluation of a simulation platform for experimentation with C/SDR. The platform developed in OPNET offers the researcher the ability to investigate how a C/SDR's settings affect metrics of interest. The platform is flexible enough to allow communication parameters to be changed on a per-packet basis. Additionally, the platform allows for the development and comparison of algorithms for configuration of a C/SDR, and forms the basis upon which future work in this area can be accomplished. Finally, it is hoped that the method used in this research will be applied to many different problems involving parametric interactions and optimization.

5.2 Findings

Even with solid theoretical and experimental analysis there were tradeoffs in achieving adequate performance over an actively jammed link. This work presented a method for balancing the tradeoffs a C/SDR must consider in meeting its goals. Using an exponential moving average in combination with the requirements vector, DOE and ANOVA techniques, this work showed that the approach is valid. Wherein, the most statistically significant parameters were adjusted dynamically to positively affect performance in the presence of a jamming node. In fact, the MCC algorithm closely matches the best static configuration in performance, while decreasing power output by 76%.

5.3 Future Work

During investigation of this thesis countless opportunities were identified for future research. The most significant of which are detailed below.

- Implement and test the method and algorithm on a hardware C/SDR platform.
- Expand the set of parameters to encompass a larger set of potential configurations.
- Consider additional environmental parameters to determine if the C/SDR can make more intelligent reconfiguration decisions when provided more information.

- Expand the set of metrics to include power consumption and power output.

- Integrate the algorithm into an online learning system, where-in historical performance data is maintained and the DOE and ANOVA are done online in response to environmental influences or changes in requirements.

Bibliography

[1] IEEE Standard 802-11. Ieee standard for wireless lan medium access control (mac) and physical layer (phy) specification, June 1997.

[2] M. Anderson and P. Whitcomb. DOE Simplified. Productivity Press, 2000.

[3] Baruch Awerbuch and Yishay Mansour. Adapting to a reliable network path. In PODC '03: Proceedings of the twenty-second annual symposium on Principles of distributed computing, pages 360–367, New York, NY, USA, 2003. ACM Press.

[4] E. Ayanoglu, S. Paul, T. F. LaPorta, K. K. Sabnani, and R. D. Gitlin. AIRMAIL: A link-layer protocol for wireless networks. ACM Wireless Networks, 1(1):47–60, 1995.

[5] H. Bai and M. Atiquzzaman. Error modeling schemes for fading channels in wireless communications: a survey. IEEE Communications Surveys, 5(2), 2003.

[6] A. Bakre and R. Badrinath. Handoff and system support for indirect tcp/ip. In Proceedings of the 2nd USENIX Symposium on Mobile and Location-Independent Computing, pages 11–24, April 1995.

[7] H. Balakrishnan, V.N. Padmanabhan, S. Seshan, and R.H. Katz. Comparing the performance of transport protocols in wireless networks. In ACM SIGCOMM, August 1996.

[8] Hari Balakrishnan, Srinivasan Seshan, Elan Amir, and Randy H. Katz. Improving tci/ip performance over wireless networks. In MobiCom '95: Proceedings of the 1st annual international conference on Mobile computing and networking, pages 2–11, New York, NY, USA, 1995. ACM Press.

[9] Chris Barrett, Achla Marathe, Madhav V. Marathe, and Martin Drozda. Characterizing the interaction between routing and mac protocols in ad-hoc networks. In MobiHoc '02: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking & computing, pages 92–103, New York, NY, USA, 2002. ACM Press.

[10] L. Berlemann, S. Mangold, and B.H.Walke. Policy-based reasoning for spectrum sharing in cognitive radio networks. In Proc. IEEE DySPAN 2005, pages 1–10, November 2005.

[11] G. Bianchi, F. Borgonovo, L. Fratta, L. Musumeci, and M. Zorzi. Cprma: a centralized packet reservation multiple access for local wireless communications. IEEE Transactions on Vehicular Technology, 46:442–446, 1997.

[12] Raffaele Bolla, Franco Davoli, and Carlo Nobile. The rra-isa multiple access protocol with and without simple priority schemes for real-time and data traffic in wireless cellular systems. MONET, 2(1):45–53, 1997.

[13] V. Bose, M. Ismert, M. Welborn, and J. Guttag. Virtual radios. IEEE/JSAC Special Issue on Software Radios, April 1999.

[14] Vanu Bose. Design and Implementation of Software Radios Using a General Purpose Processor. PhD thesis, MIT, 1999.

[15] Bob Brewin. Jtrs costs leap higher. Federal Communication Weekly, May 2005.

[16] Kevin Brown and Suresh Singh. M-tcp: Tcp for mobile cellular networks. SIGCOMM Comput. Commun. Rev., 27(5):19–43, 1997.

[17] M.M. Buddhikot, P. Kolodzy, S. Miller, K. Ryan, and J. Evans. Dimsumnet: new directions in wireless networking using coordinated dynamic spectrum access. Proc. IEEE WoWMoM 2005, 2005.

[18] C.G. Christodoulou. Rf-mems and its applications to microwave systems, antennas and wireless communications. Microwave and Optoelectronics Conference, 1:525–531, 2003.

[19] Federal Communications Commission. Fcc spectrum policy task force report. Technical Report ET Docket No. 02- 135, FCC, November 2002.

[20] Federal Communications Commission. Establishment of an interference temperature metric to quantify and manage interference and to expand available unlicensed operation in certain fixed, mobile and satellite frequency bands. ET Docket No. 03-237 FCC 03-289, FCC, November 2003.

[21] Federal Communications Commission. Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies. ET Docket No. 03-108 FCC 03-322, FCC, December 2003.

[22] Federal Communications Commission. Facilitating opportunities for flexible, efficient, and reliable spectrum use employing cognitive radio technologies. ET Docket No. 03-108 FCC 05-57, FCC, March 2005.

[23] C. Cordeiro, K. Challapali, D. Birru, and S. Shankar. Ieee 802.22: the first worldwide wireless standard based on cognitive radios. In Proc. IEEE DySPAN 2005, pages 328–337, November 2005.

[24] George Coulouris, Jean Dollimore, and Tim Kindberg. Distributed Systems Concepts and Designs. Addison-Wesley, 2001.

[25] Christian Doerr, Michael Neufeld, Jeff Fifield, Troy Weingart, Douglas C. Sicker, and Dirk Grunwald. Multimac - an adaptive mac framework for dynamic radio networking. In Dynamic Specturm Allocation Networks, 2005.

[26] David A. Eckhardt and Peter Steenkiste. Improving wireless LAN performance via adaptive local error control. Proceedings of IEEE ICNP '98, 1998.

[27] P. Elias. Coding for noisy channels. IRE Conv, 4:37–47, 1955.

[28] A. Faragó, A.D. Myers, V.R. Syrotiuk, and G.V. Záruba. Meta-mac protocols: Automatic combination of mac protocols to optimize performance for unknown conditions. EEE JSAC, special issue on: Analysis and Synthesis of MAC Protocols, 18:1670–1682, 2000.

[29] A. Faragó, A.D. Myers, V.R. Syrotiuk, and G.V. Záruba. A new approach to mac protocol optimization. In Proc. IEEE GLOBECOM'01, 2001.

[30] Lawrence L. Garlick, Raphael Rom, and Jonathan B. Postel. Reliable host-to-host protocols: Problems and techniques. In SIGCOMM '77: Proceedings of the fifth symposium on Data communications, pages 4.58–4.65, New York, NY, USA, 1977. ACM Press.

[31] A. Goldsmith and S. Wicker. Design challenges for energy-constrained ad hoc wireless networks. IEEE Wireless Communications, 9:8–27, 2002.

[32] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard, and B. Ramamurthi. Packet reservation multiple access for local wireless communications. IEEE Trans. Commun., 37:885–890, 1989.

[33] Zygmunt J. Haas and Prathima Agrawal. Mobile-TCP: An asymmetric transport protocol design for mobile systems. In ICC (2), pages 1054–1058, 1997.

[34] R. Hamming. Error detecting and error correcting codes. Bell Systems Tech. Journal, 29:147–160, 1950.

[35] S. Haykin. Cognitive radio: brain-empowered wireless communications. IEEE Journal on Selected Areas in Communications, 23:201–220, 2005.

[36] William D. Horne. Adaptive spectrum access: Using the full spectrum space. In Telecommunications Policy Research Conference, 2003.

[37] Christian Huitema. Routing in the Internet. Prentice Hall PTR, 2000.

[38] Pierre A. Humblet and Stuart R. Soloway. A fail-safe layer for distributed network algorithms and changing topologies. Submitted to IEEE Trans. Comm., 1987.

[39] Joseph Mitola III. Cognitive Radio An Integrated Agent Architecture for Software Defined Radio. PhD thesis, Royal Institute of Technology (KTH), 2000.

[40] ITU. Wp8a draft report on sdr, September 2004.

[41] R. Jiang, V. Gupta, and C. Ravishankar. Interactions between tcp and the ieee 802.11 mac protocol. In Proc. of DISCEX, 2003.

[42] David B. Johnson. Scalable support for transparent mobile host internetworking. Wireless Networks, 1(3):311–321, 1995.

[43] Phil Karn. Maca—a new channel access method for packet radio. In Proceedings of 9th Amateur Radio Computer Networking Conference, 1990.

[44] Mark J. Karol, Kai Y. Eng, and Zhao Liu. An efficient demand-assignment multiple access protocol for wireless packet (atm) networks. Wireless Networks, 1(3):269–279, 1995.

[45] Vikas Kawadia and P. R. Kumar. A cautionary perspective on cross layer design. IEEE Wireless Communication Magazine, 12:3–11, 2005.

[46] Young-Bae Ko, Vinaychandra Shankarkumar, and Nitin H. Vaidya. Medium access control protocols using directional antennas in ad hoc networks. In INFOCOM (1), pages 13–21, 2000.

[47] James F. Kurose and Keith Ross. Computer Networking: A Top-Down Approach Featuring the Internet. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003.

[48] Raymond J. Lackey and Donal W. Upmal. Speakeasy: The military software radio. IEEE Communications Magazine, 33(5):56–61, May 1995.

[49] Paul Lettieri and Mani B. Srivastava. Adaptive frame length control for improving wireless link throughput, range and energy efficiency. In INFOCOM (2), pages 564–571, 1998.

[50] S. Lin and D. J. Costello Jr. Error Control Coding: Fundamentals and Applications. Prentice Hall, 1983.

[51] Reiner Ludwig, Almudena Konrad, Anthony D. Joseph, and Randy H. Katz. Optimizing the end-to-end performance of reliable flows over wireless links. Wireless Networks, 8(2/3):289–299, 2002.

[52] Reiner Ludwig, Bela Rathonyi, Almudena Konrad, Kimberly Oden, and Anthony Joseph. Multi-layer tracing of tcp over a reliable wireless link. SIGMETRICS Perform. Eval. Rev., 27(1):144–154, 1999.

[53] Matthew Mathis and Jamshid Mahdavi. Forward acknowledgement: Refining TCP congestion control. In SIGCOMM, pages 281–291, 1996.

[54] R. M. Metcalf and D. R. Boggs. Ethernet: Distributed packet switching for local computer networks. Communications of the ACM, 19(7):395–404, July 1976.

[55] J. Mikkonen, J. Aldis, G. Awater, A.S. Lunn, and David Hutchison. The magic wand - functional overview. IEEE Journal on Selected Areas in Communications, 16(6):953–972, August 1998.

[56] Andrew D. Myers, Gergely V. Záruba, and Violet R. Syrotiuk. An adaptive generalized transmission protocol for ad hoc networks. Mob. Netw. Appl., 7(6):493–502, 2002.

[57] Narasimhan and R.D. Yates. A new protocol for the integration of voice and data over prma. IEEE JSAC, 14(4):623–631, October 1995.

[58] J. Neel, R.M. Buehrer, B.H. Reed, and R.P Gilles. Game theoretic analysis of a network of cognitive radios. Circuits and Systems, 3:409–412, 2002.

[59] J. Neel, R.M. Buehrer, B.H. Reed, and R.P Gilles. Convergence of cognitive radio networks. Wireless Communications and Networking Conference, 4:2250–2255, 2004.

[60] James O. Neel, Jeffrey H. Reed, and Robert P. Gilles. Game models for cognitive radio algorithm analysis. In Software Defined Radio Forum Technical Conference, pages 27–32, 2004.

[61] Michael Neufeld, Jeff Fifield, Christian Doerr, Anmol Sheth, and Dirk Grunwald. Softmac flexible wireless research platform. In HOTNETS, 2005.

[62] S.M. Nishra, D. Cabric, C. Chang, D. Willkomm, B. Schewick, A. Wolisz, and R.W. Brodersen. A real time cognitive radio testbed for physical and link layer experiments. In Proc. IEEE DySPAN 2005, pages 562–567, November 2005.

[63] D. Petras. Performance evaluation of medium access control protocols for mobile broadband systems. Technical report, Technical Report RACE project, January 1996.

[64] Allen Petrin and Paul G. Steffes. Analysis and comparison of spectrum measurements performed in urban and rural areas to determine the total amount of spectrum usage. In Proceedings of the International Symposium on Advanced Radio Technologies, pages 9–12, March 2005.

[65] E. Prange. Cyclic error-correcting codes in two symbols. Technical Report AFCRC-TN-57, Air Force Cambridge Research Center, 1957.

[66] Ram Ramanathan. On the performance of ad hoc networks with beamforming antennas. In MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing, pages 95–105, New York, NY, USA, 2001. ACM Press.

[67] S. Ramanathan and Martha Steenstrup. A survey of routing techniques for mobile communications networks. Mobile Networking Applications, 1(2):89–104, 1996.

[68] Ananth Rao and Ion Stoica. An overlay mac layer for 802.11 networks. In MobiSys '05: Proceedings of the 3rd international conference on Mobile systems, applications, and services, pages 135–148, New York, NY, USA, 2005. ACM Press.

[69] I. Reed. A class of multiple-error-correcting codes and a decoding scheme. IEEE Transactions on Information Theory, 4:38–49, 1954.

[70] Christian Rieser. Biologically Inspired Cognitive Radio Engine Model Utilizing Distributed Genetic Algorithms for Secure and Robust Wireless Communications and Networking. PhD thesis, Virginia Polytechnic Institute and State University, August 2004.

[71] A. Sahai, N. Hoven, and R. Tandra. Some fundamental limits in cognitive radio. In Allerton Conference on Communications, Control and Computing, October 2004.

[72] Ricardo J. Sánchez, Joseph B. Evans, Gary J. Minden, Victor S. Frost, and K. Sam Shanmugan. Rdrn: a prototype for a rapidly deployable radio network. SIGMOBILE Mobile Computing Communications Review, 2(2):15–22, 1998.

[73] B. A. Sharp, E. A. Grindrond, and D.A. Camm. Hybrid tdma/csma protocol for self managing packet radio networks. In IEEE ICUPC, pages 929–933, 1995.

[74] Peerapon Siripongwutikorn, Sujata Banerjee, and David Tipper. A survey of adaptive bandwidth control algorithms. IEEE Communications Surveys, 5(1), 2003.

[75] Andrew Tanenbaum. Computer Networks. Prentice Hall Professional Technical Reference, 2002.

[76] David L. Tennenhouse and Vanu G. Bose. Spectrumware: a software-oriented approach to wireless signal processing. In MobiCom '95: Proceedings of the 1st annual international conference on Mobile computing and networking, pages 37–47, New York, NY, USA, 1995. ACM Press.

[77]] R.W. Thomas, L.A. DaSilva, and A.B. MacKenzie. Cognitive networks. In Proc. IEEE DySPAN 2005, pages 352–360, November 2005.

[78] K. Vadde and V. Syrotiuk. Quantifying factors affecting quality of service in mobile ad hoc networks. SIMULATION, 81(8):547–560, 2005.

[79] K. K. Vadde and V. R. Syrotiuk. Factor interaction on service delivery in mobile ad hoc networks. IEEE Journal on Selected Areas in Communications, 22:1335–1346, September 2004.

[80] Kiran K. Vadde, Member-Violet R. Syrotiuk, and Douglas C. Montgomery. Optimizing protocol interaction using response surface methodology. IEEE Transactions on Mobile Computing, 5(6):627–639, 2006.

[81] Website. cran.r-project.org.

[82] Website. e2r2.motlabs.com.

[83] Website. jtrs.army.mil.

[84] Website. standards.ieee.org /getieee802 /portfolio.html.

[85] Website. www.altera.com.

[86] Website. www.darpa.mil/ato/programs/xg/.

[87] Website. www.investopedia.com.

[88] Website. www.ofcom.org.uk.

[89] Website. www.opnet.com.

[90] Website. www.statease.com.

[91] Website. www.vanu.com.

[92] Troy Weingart. A Method for Dynamic Reconfiguration of a Cognitive Radio System. PhD thesis, University of Colorado at Boulder, December 2006.

[93] Troy Weingart, Doug Sicker, Dirk Grunwald, and Michael Neufeld. Adverbs and adjectives: An abstraction for software defined radio. In Proceedings of the International Symposium on Advanced Radio Technologies, pages 183–192, March 2005.

[94] Troy Weingart, Douglas C. Sicker, and Dirk Grunwald. Evaluation of cross-layer interactions for reconfigurable radio platforms. In Technology and Policy for Accessing Spectrum (TAPAS), 2006.

[95] Troy Weingart, Douglas C. Sicker, and Dirk Grunwald. A method for dynamic configuration of a cognitive radio. In Networking Technologies for Software Defined Radio (SDR) Networks, 2006.

[96] Troy Weingart, Douglas C. Sicker, and Dirk Grunwald. A predictive model for cognitive radio. In Military Communications (MILCOM), 2006.

[97] Rhonda Whickham. Long-awaited sdr becomes commercial reality. Wireless Week, page 62, March 2005.

[98] D. Willkomm, J. Gross, and A. Wolisz. Reliable link maintenance in cognitive radio systems. In Proc. IEEE DySPAN 2005, pages 371–378, November 2005.

[99] W.C. Wong and D.J. Goodman. Integrated data and speech transmission using packet reservation multiple access. In IEE ICC, pages 172–176, May 1993.

[100] T. S. Yum and K. W. Hung. Design algorithms for multihop packet radio networks with multiple directional antennas stations. IEEE Transactions on communications, 40:1716–1724, 1992.

[101] M. Zorzi and R. Rao. Energy constrained error control for wireless channels. IEEE Personal Communications, 4:27–33, December 1997.